AFRL-IF-RS-TR-2005-147
Final Technical Report
April 2005

MULTIATTRIBUTE UTILITY ANALYSIS FOR ULTRALOG

Decision Science Associates, Incorporated

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J777

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-147 has been reviewed and is approved for publication

APPROVED: /s/

DEBORAH A. CERINO
Project Engineer

FOR THE DIRECTOR: /s/

JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
APRIL 2005

3. REPORT TYPE AND DATES COVERED
Final Apr 00 – Dec 04

4. TITLE AND SUBTITLE
MULTIATTRIBUTE UTILITY ANALYSIS FOR ULTRALOG

5. FUNDING NUMBERS
C - F30602-00-C-0089
PE - 62702E
PR - IAST
TA - 00
WU - 10

6. AUTHOR(S)
Jacob W. Ulvila and
James O. Chinnis, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Decision Science Associates, Incorporated
PO Box 969
Vienna Virginia 22183-0969

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFED
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2005-147

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Deborah A. Cerino/IFED/(315) 330-1445/ Deborah.Cerino@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

The goal of this project was to develop and apply appropriate metrics to represent the functional and survivability characteristics of an experimental agent-based logistics system (UltraLog) and to apply the metrics in tools that would assist system designers, developers, and evaluators to compare, quantify, generate figures of merit, and make decisions. To make a survivability claim for UltraLog, it was necessary to demonstrate a particular level of survivability of UltraLog's performance in the face of a given level of information system infrastructure loss. Performance was quantified using concepts from multiattribute utility theory and model parameters elicited from groups of experts. Infrastructure loss was quantified in terms of a novel approach partially based on information system path complexity.

14. SUBJECT TERMS
Multiattribute Utility Theory, Logistic, Infrastructure, Metrics, Decision Analysis

15. NUMBER OF PAGES
105

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

LIST OF TABLES

EXECUTIVE SUMMARY

The work performed under this contract in support of DARPA's UltraLog program is described in this technical report. It primarily describes ways that were developed and applied to quantify the survivability, capability and performance of UltraLog. It discusses how multiattribute utility (MAU) analysis can be and was used to support assessment and design decisions for UltraLog. In addition, it describes the approaches considered and developed for the quantification of information system infrastructure loss.

The graduation objective for the UltraLog Program is defined as 1000 medium-complexity agents operating under directed adversary attack with up to 45% information infrastructure loss operating in complex, dynamic environments with not more than 20% capabilities degradation and not more than 30% performance degradation for a period representing 180 days of sustained military operations in a major regional contingency. Establishing that UltraLog achieves this level of capability and performance in the face of a particular level of infrastructure loss is referred to as making the "survivability claim."

The work reported here primarily addressed the issues of how to measure the "capabilities degradation," the "performance degradation," and the "infrastructure loss" associated with any particular experimental test under applied stresses. Without measuring those variables, no survivability claim could be made.

This report primarily contains: 1) a general description of multiattribute utility (MAU) methods, 2) descriptions of MAU methods developed for addressing important UltraLog assessment and design decisions, 3) a user's guide to the prototype software tool, MAUL (MultiAttribute utility for UltraLog), 4) a discussion of issues related to the quantification of infrastructure loss, and 5) a discussion of the selected approach to infrastructure loss.

The general MAU-related description includes the topics of: identifying attributes, establishing utility functions, establishing tradeoffs, assessing the performance of systems or system designs against attributes, calculating results, and sensitivity analyses. This description also addresses extensions to the basic method that deal with: interdependence in the values of attributes; interdependence in the technical performance of alternatives against attributes; goals, requirements, and performance thresholds; and risk, uncertainty, and attitude toward risk taking.

The description of the application of the MAU methodology to UltraLog illustrates how the methods should be applied to evaluate UltraLog design and development alternatives. This includes illustrations of how to address such questions as: *What is the evaluation of the current UltraLog development? How does this compare with the requirements and goals? How does this compare with the baseline? How are the evaluations changing over time? What attributes need to be enhanced to improve the evaluation?* and *Which areas of enhancement offer the greatest opportunities for improvements?*

The first description of the application of MAU shows how MAU serves as a unifying methodology for UltraLog assessments. MAU provides the basis for quantitative functional assessments, security assessments, robustness assessments, and scalability assessments. It provides the basis to combine a security assessment, robustness assessment, and scalability assessment into a survivability assessment or a functional assessment, survivability assessment, and cost assessment into an overall assessment. MAU provides a single, overall figure of merit for an UltraLog system and for any of its component attributes such as functional, security, robustness, and scalability attributes. Finally, MAU provides the method for making measurable assessments of an UltraLog system against its goals.

Detailed illustrations are provided for these component analyses. A complete illustrative functional assessment is described. It illustrates the six steps of an MAU analysis for the two functional measures of effectiveness (MOEs), six operational impacts (OIs), and thirty-two measures of performance (MOPs). This includes: 1) structuring the MOEs, OIs, and MOPs in an MAU hierarchy; 2) specifying metrics and utility functions for the MOPs; 3) establishing weights across MOEs, OIs, and MOPs; 4) assessing the performance of UltraLog systems against the attributes; 5) calculating the functional assessment; and 6) performing sensitivity analyses.

The illustrative security assessment starts with a description of six security Red Team flags as attributes. The illustration then describes eleven other security attributes that should be considered in a security assessment of UltraLog. The first five steps in an MAU analysis are then illustrated for the Red Team portion of a security assessment of UltraLog.

The remaining illustrations show how MAU analyses address quantitative statements in the UltraLog survivability claim and how the results from those analyses could be used in an assessment of robustness. This report demonstrates how the determination of information infrastructure loss depends on the information infrastructure assets and their configuration and provides a method for determining the percentage information infrastructure loss. Several different approaches are discussed, followed by a detailed discussion of the approach ultimately adopted for UltraLog.

MAU methods are used to illustrate how they relate to quantitative assessments of capabilities degradation and performance degradation starting by identifying attributes of capability and attributes of performance. The functional MOPs are grouped into the categories of capability and performance. Completing the steps of the MAU analysis provides quantitative measures (utilities) of the capability and performance of a given UltraLog system, a baseline system, and a hypothetical system that meets all goals. These utilities can be used to calculate capability and performance degradation from either the goal or the baseline.

A spreadsheet-based tool, MAUL (MultiAttribute utility for UltraLog), was developed as part of this research and was delivered as a set of Excel spreadsheet files and algorithm descriptions. This report contains a guide for using this tool, which became a part of the UltraLog software that built the experimental databases. MAUL is implemented as an Excel spreadsheet with eleven tabs. It supports the six steps of an MAU analysis described in the report. Separate files were delivered for the five MAU analyses described in the report: functional assessment, Red Team security assessment, capabilities degradation, performance degradation, and robustness assessment. The report describes the operation of each tab. By following the instructions, a user is able to modify the inputs to any of the five illustrative MAU analyses or to develop a new MAU analysis from scratch.

After the approaches, tools, and illustrative materials were developed and pre-tested and revised, the required utility functions and weights were assessed in structured meetings with experts. These efforts sometimes resulted in it becoming apparent that changes were needed in the hierarchies of MAU attributes themselves. Those changes were made during the course of the elicitations. A near-final hierarchy and a set of utility curves and weights was determined during multiday group meetings. A number of proven elicitation methods were employed, including “balance beam” for paired-comparisons, “pricing out,” and direct elicitation of “swing weights.”

In order to make the survivability claim, it was necessary not just to measure performance and capability using MAU; infrastructure loss had to be measured as well. A number of different approaches were developed and analyzed. Essentially, these attempted to develop a reasonable metric for infrastructure loss based on losses suffered in specific processors and connectors, given a network configuration. The approach selected and used in the UltraLog evaluation centered on the notion of path complexity. A spreadsheet based tool was developed for experiment planning, and the algorithms were transferred to other contractors for inclusion in the experimental software so that infrastructure loss could be included in the experiment databases.

MAU analysis as described in this report provided a unifying methodology to combine the efforts of the UltraLog contractors. For example, a complete specification of an MAU analysis provided overall direction to the contractors by showing: quantitative goals for performance; tradeoffs within and among attributes; and the relative importance of deficiencies discovered during experiments and tests. Further, it provided the rigorous basis for the measurement of survivability, capability, and performance, which—along with the basis for measuring infrastructure loss—provided the analytical basis for the survivability claim.

1. INTRODUCTION

This section presents an overview of the work performed on the DARPA project, “IA Metrics Decision Support Tool,” and provides a description of the organization of the report. This report describes accomplishments during the period February 2001 through December 2004. During the initial eight months of that period, the research was supported by close-out funds from DARPA’s IASET Program, but technical direction came from the UltraLog Program. Both funds and direction for the remainder of the contract period were from the UltraLog Program. There were two principal foci of the effort reported here. The first was the development and application of multiattribute utility (MAU) methods for UltraLog that could provide the needed measures of performance and survivability. MAU analysis as described in this report provides a unifying methodology to combine the efforts of the UltraLog contractors. The second was the development and application of methods for the quantification of the infrastructure loss sustained in an information system such as UltraLog. In both cases, spreadsheet-based tools were provided for planning purposes and the concepts were reduced to algorithms for incorporation by other UltraLog contractors into the software that produced the experimental data.

1.1 Overview

The goal of this project was to develop appropriate metrics to represent the functional and survivability characteristics of UltraLog and to apply the metrics in tools that would assist system designers, developers, and evaluators to compare, quantify, generate figures of merit, and make decisions.

To make a survivability claim for UltraLog, it was necessary to demonstrate a particular level of survivability of UltraLog’s *performance* in the face of a given level of information system *infrastructure loss*. Performance could be quantified using concepts from MAU. The other measurement required to make the claim of UltraLog survivability is that of infrastructure. By “infrastructure” is meant the information system (IS) substrate upon which UltraLog and its agents reside. In any case, “infrastructure loss” was an imprecise term that was made precise and quantifiable.

Performance was quantified by using MAU. MAU is a method for performing evaluations with multiple objectives, criteria, or impacts. MAU provides a mathematically sound procedure for assessing one’s preferences over multiple attributes. MAU models reflect explicitly the relative importance of attributes, tradeoffs among attributes, and the degree to which system designs or developments impact attributes. MAU methods can support system designers and developers in coming to a consensus about the importance of attributes, and MAU supports the recording and recovery of the rationale behind evaluations.

This report describes, among other things, how MAU analysis was used to support assessment and design decisions for UltraLog. It contains: 1) a general description of MAU methods, 2) descriptions of MAU methods to address important UltraLog assessment and design decisions, and 3) a user’s guide to a spreadsheet-based tool, MAUL (MultiAttribute utility for UltraLog).

The general description includes the topics of: identifying attributes, establishing utility functions, establishing tradeoffs, assessing the performance of systems or system designs against attributes, calculating results, and sensitivity analyses. This description also addresses extensions to the basic method that deal with: interdependence in the values of attributes; interdependence in the technical performance of alternatives against attributes; goals, requirements, and performance thresholds; and risk, uncertainty, and attitude toward risk taking.

The description of the application of the MAU methodology to UltraLog describes how the methods were applied to evaluate UltraLog design and development alternatives, to support experiments, and to support the overall claim. The discussion covers how MAU defined metrics for the goal, provided metrics for experiments, guided the development of UltraLog, tracked progress towards the goal, and provided the basis for the "Graduation Claim."

A spreadsheet-based tool, MAUL, was developed as part of this research. MAUL supports the six steps of an MAU analysis described in the report. The prototype tool demonstrated the feasibility of an MAU analysis tool and the approach was incorporated by other Ultralog contractors into the software that produced the experimental data.

In order to make the survivability claim, it was necessary not just to measure performance and capability using MAU; infrastructure loss had to be measured as well. A number of different approaches were developed and analyzed. Essentially, these attempted to develop a reasonable metric for infrastructure loss based on losses suffered in specific processors and connectors, given a network configuration. The approach selected and used in the UltraLog evaluation centered on the notion of path complexity. A spreadsheet based tool was developed for experiment planning, and the algorithms were transferred to other contractors for inclusion in the experimental software so that infrastructure loss could be included in the experiment databases.

MAU analysis as described in this report provided a unifying methodology to combine the efforts of the UltraLog contractors. For example, a complete specification of an MAU analysis provided overall direction to the contractors by showing: quantitative goals for performance; tradeoffs within and among attributes; and the relative importance of deficiencies discovered during experiments and tests. Further, it provided the rigorous basis for the measurement of survivability, capability, and performance, which—along with the basis for measuring infrastructure loss—provided the analytical basis for the survivability claim.

1.2 Report Organization

This report is organized as follows. Section 1 presents an overview of the project and describes the organization of the report.

Section 2 contains a description of multiattribute utility (MAU) methods. This includes descriptions of how to: determine the MAU analysis structure of attributes, establish utility functions, establish tradeoffs, calculate results, and conduct sensitivity analyses. Section 2 also describes extensions to the basic method to deal with probability and uncertainty, interdependencies, and attitude toward risk.

Section 3 describes how MAU was provided as a unifying methodology for UltraLog assessments. It shows the top level of an initial MAU analysis for UltraLog with attributes of: functional assessment, cost, and survivability (which comprises security, robustness, and scalability).

Section 4 contains a complete, illustrative MAU functional assessment. It illustrates the six steps of an MAU analysis for the two functional measures of effectiveness (MOEs), six operational impacts (OIs), and thirty-two measures of performance (MOPs). This includes: 1) structuring the MOEs, OIs, and MOPs in an MAU hierarchy; 2) specifying metrics and utility functions for the MOPs; 3) establishing weights across MOEs, OIs, and MOPs; 4) assessing the performance of UltraLog systems against the attributes; 5) calculating the functional assessment; and 6) performing sensitivity analyses.

Section 5 contains an illustrative security assessment. It starts with a description of six security Red Team flags as attributes. It then describes eleven other security attributes that should be considered in a security assessment of UltraLog. The first five steps in an MAU analysis are then illustrated for the Red Team portion of a security assessment of UltraLog.

Section 6 summarizes work performed on the elicitation of utility curves and weights and revisions to the model structure. After the approaches and illustrative materials described in Sections 4 and 5 were developed and pre-tested and revised, the required utility functions and weights were assessed in structured meetings with experts. These efforts sometimes resulted in it becoming apparent that changes were needed in the hierarchies of attributes themselves. Those changes were made during the course of the elicitations. A near-final hierarchy and a set of utility curves and weights was determined during multiday group meetings.

Section 7 contains a method for determining information infrastructure loss, with particular attention to the determination of "45% information infrastructure loss." It demonstrates how the determination of information infrastructure loss depends on the information infrastructure assets and their configuration. The method is illustrated in a simple configuration of information infrastructure and under three different cases that vary the principles for determining when infrastructure is considered lost and the information infrastructure value of different classes of assets. In all cases, the analysis shows that many combinations of lost assets could result in the same information infrastructure loss, especially near 45% loss. Extensions to address partial loss of information infrastructure and other variations are also discussed.

Section 8 describes further work on the measurement of infrastructure loss, based on the concept of path complexity. As the nature of the UltraLog system became better defined in the course of the project, the idea arose of basing information system infrastructure on path complexity, as well as the viability of processors and connections. That distinct approach was developed and was the version selected for use in the assessments.

Section 9 describes an MAU method for determining quantitative assessments of capabilities degradation and performance degradation. This addresses the part of the UltraLog goal that refers to "not more than 20% capabilities degradation and not more than 30% performance degradation." The method starts by identifying attributes of capability and attributes of performance. As an illustration, the functional MOPs are grouped into the categories of capability and performance. Completing the steps of the MAU analysis provides quantitative measures (utilities) of the capability and performance of a given UltraLog system, a baseline system, and a hypothetical system that meets all goals. These utilities can be used to calculate capability and performance degradation from either the goal or the baseline.

Section 10 contains an illustrative MAU analysis of robustness. Robustness might be defined as the property of UltraLog of minimizing the effects of the causes of variation without eliminating the causes. As related to the goal, UltraLog might be considered robust if it is able to: "operate with up to 45% information infrastructure loss in a very chaotic environment with not more than 20% capabilities degradation and not more than 30% performance degradation." The report illustrates how the degree of robustness could be determined from an MAU analysis that evaluates capability and performance degradation under conditions of information infrastructure loss and chaos in the environment.

Section 11 contains what is essentially a user's guide for MAUL (MultiAttribute utility for UltraLog), the spreadsheet-based tool that was developed as part of this research. MAUL is implemented as an Excel spreadsheet with eleven tabs. It supports the six steps of an MAU analysis. The user's guide describes how MAUL can be used to support all six steps of an MAU analysis. It contains illustrations from the MAU functional assessment.

Section 12 discusses how the MAU and infrastructure loss metrics developed and discussed in this report formed the basis for using the experimental data to make the survivability claim.

References are provided between Section 9 and the Appendices.

Appendix A contains the preliminary UltraLog functional assessment developed by Los Alamos Technical Associates.

2. MULTIATTRIBUTE UTILITY (MAU) METHODS

A major goal of this contract was to develop appropriate metrics to represent the functional and survivability characteristics of UltraLog and to apply the metrics in a decision support tool that will assist system designers, developers, and evaluators to compare, quantify, generate figures of merit, and make decisions.

The basis for the methodology is multiattribute utility (MAU) analysis, a method for performing evaluations with multiple objectives, criteria, or impacts. MAU provides a mathematically sound procedure for assessing one's preferences over multiple attributes. MAU models reflect explicitly the relative importance of attributes, tradeoffs among attributes, and the degree to which system designs or developments impact attributes. MAU methods can support system designers and developers in coming to a consensus about the importance of attributes, and MAU supports the recording and recovery of the rationale behind evaluations.

MAU is described in detail by Keeney and Raiffa (1976). As applied here, MAU analysis involves six steps: structuring attributes, establishing utility functions, specifying tradeoffs, assessing designs, calculating results, and performing sensitivity analyses. Section 2 describes MAU in general terms and indicates ways that MAU was applied to UltraLog.

2.1 Determine the Multiattribute Utility (MAU) Analysis Structure of Attributes

The first step an MAU analysis of UltraLog is to develop a framework that encompasses all important attributes of UltraLog, the metrics associated with the attributes, and the interrelationships among attributes. This provides a common frame of reference to foster common understanding among designers, developers, evaluators, and managers.

In an MAU model structure, attributes are specified in a hierarchical fashion with the most general ones at the top and most specific ones at the bottom. For UltraLog, the top level of the hierarchy would contain survivability attributes and the functional and cost attributes that might have to be traded off against the survivability attributes. The hierarchical MAU structure shows the relationships among the attributes, and it provides a trace or audit trail from an evaluation to the attributes. Figure 3-1 in the next section of this report shows an example of an MAU hierarchy of attributes for UltraLog value. The top level shows that UltraLog value consists of functional attributes, cost attributes, and survivability attributes, each of which is divided into subattributes. Survivability is divided into Scalability, Robustness, and Security. Each subattribute is divided further. For example, Security is divided into attributes that can be determined by Red Team evaluations and those that cannot be so determined. Red Team attributes are the flags that the Red Team pursues in the evaluations.

The location of an attribute in the hierarchy does not imply anything about its importance. Importance is determined by the method described in Section 2.3. This could result in an attribute that appears at the bottom of the hierarchy being the most important. Nor does the location of an attribute affect which tradeoffs are made among attributes. All attributes are traded off, explicitly or implicitly, against all others. The hierarchy provides a convenient way to describe attributes and their relationships (e.g., if one attribute comprises several subattributes). An MAU hierarchy is not unique, and several different hierarchies may be equally adequate to describe the attributes in any given MAU analysis.

Attributes should reflect the full range of considerations that are important to the evaluation, not just the ones that are easy to measure. In particular, subjective criteria and qualitative attributes can be included if they can be defined clearly. In developing the MAU structure, one seeks a set of attributes that are: comprehensive enough to account for most of what is important; able to highlight the differences among the items being evaluated (e.g., different years' versions of the UltraLog implementations or alternative designs); reliable and reproducible; intelligible, in order to facilitate understanding and enhance defensibility; and reflect separate, non-overlapping values, to avoid double counting. One also seeks, to the extent possible, to structure attributes that are additive *in preference*. Loosely speaking, attributes are additive in preference if one's preferences over an attribute do not depend on the levels of the other attributes. This is not always possible, in which case a non-additive structure is required. See section 2.5 for a discussion of ways that non-additivity can be accommodated in the method.

2.2 Establish Utility Functions

The second step in the development of an MAU analysis is to establish a utility function for each attribute. The utility function describes how much the decision maker cares about changes in the attribute. A utility function is designed to represent on a ratio scale the value of different amounts of the attribute. A variety of techniques may be used to develop a utility function. A utility function may be an absolute scale against a measurable unit or it may be a relative scale defined by reference to qualitative descriptors. Absolute scales are more common for attributes for which metrics have been developed. Relative scales are more common for subjective or qualitative attributes. Utility is unique up to a linear transformation, and it is usually convenient to specify a utility function on a common, continuous ratio scale of value (i.e., fractional values are allowed) to permit later comparisons among attributes. A numerical scale of preference offers the most compact and precise means of expressing value. The ratio nature of the scale is an important feature that justifies the mathematical techniques used in MAU. An interpretation of the ratio scale is that a change in score from 0 to 50 is half as important as a change from 0 to 100. Carefully developed ratio scales ensure accurate statements about and assessments of value that encompass both objective and subjective attributes in a consistent and comparable manner.

Ratio scales result in different possible shapes for curves that relate metric units to the utility scale. For example, shapes can be linear (increasing or decreasing), a step function, a curve of increasing or decreasing slope, S-shaped, U-shaped (normal or upside down), continuous or discrete (e.g., for a categorical variable).

For each attribute and metric in the structure, a quantitative utility scale is developed and, where appropriate, qualitative descriptors defined. The process is designed so as not to lose information contained in qualitative descriptors. It always maintains an "audit trail" to the qualitative description even after utility scales are defined. This allows a decision maker to trace an evaluation to the qualitative descriptors.

The utility scales described here are *value* scales, and value is an inherently subjective concept. The elicitation of value requires special techniques, and a number of different techniques might be used. These include ranking and rating, paired comparison, and mathematical formulas.

2.3 Establish Tradeoffs and Calculate Results

The final steps in an MAU analysis are to establish tradeoffs and calculate results. The utility scales are developed separately for each attribute and are not comparable between attributes. To make comparisons between attributes and to reflect differences in importance of different attributes, a set of weights is assessed. The common perception of a weight is that it answers the question, "How important is attribute A relative to attribute B?" As Keeney (1992) notes, this "is one mistake that is very commonly made in prioritizing objectives. Unfortunately, this mistake is sometimes the basis for poor decision making. It is always a basis for poor information." The problem is that the question does not compare the ranges of improvement represented on the attribute. The more pertinent question is, "How important is the difference in values for attribute A compared with the difference for attribute B?" These are known as swing weights. As an example, suppose you were evaluating two automobiles on the basis of cost and fuel economy. You might state that fuel economy is twice as important as cost. However, the response should depend on the ranges of values considered. For example, a range from 26 miles per gallon to 27 mpg should be less important than a range from 9 to 35 mpg (with the range on cost held constant). MAU analyses include ratio judgments of swing weights at each level in the hierarchy. Swing weighs are an important part of the mathematical basis for MAU. At each node in the hierarchy, swing weights are assessed on a ratio scale. They are then normalized to sum to 1.0 for ease of comparison, again on a continuous scale. Swing weights in the hierarchical MAU structure might be assessed top-down or bottom-up. These two methods produce the same results, after applying the proper consistency checks. The assessment procedure for weights should produce the same contribution for a given attribute regardless of where that attribute happens to be placed in the hierarchy. It is the importance of the swing in the attribute that determines how much it contributes to the analysis, not its position in the hierarchy. Furthermore, all attributes are traded off, implicitly or explicitly, against all others.

A number of different techniques may be used to elicit swing weights. These include:

Paired comparisons—comparisons between selected individual pairs of attributes or between selected pairs of groups of attributes that indicate <, =, or > relationships, which provide the information needed to assign a set of weights that add to 1.0.

Reference comparison—attribute swings are compared with a “reference attribute.” The reference attribute is assigned a value of 100 arbitrarily and other attributes are assigned values to represent the importance of their swings relative to the 100. After all values have been assigned, they are normalized.

Distributing 100 points—distributing 100 points among the attributes (normalization is achieved by dividing the distributions by 100).

This procedure for assigning weights is appropriate for additive independent preferences. More complicated procedures, described by Keeney and Raiffa (1976), might be used if values are not additive. Some particular non-additive formulations are discussed in Section 2.5. All procedures require preferences to obey elementary rules of logic, such as transitivity—i.e., if A>B and B>C, then A>C. Checks can be built into a decision support tool that will alert the user to inconsistencies as they occur.

Next, the performance of a system or system design is assessed against attributes. Assessments could incorporate results from analytical evaluations, experiments, engineering judgment, or other forms of evidence. In the case of UltraLog, we expect that some assessments will come out of the annual experiments and Red Team evaluations and others will come from separate assessments of experts in logistics, computer security, and other relevant fields. These assessments are translated into utilities using the utility functions.

With an additive MAU structure, a system’s or design’s evaluation is calculated as a weighted average utility. This is determined by taking a system’s or design’s utilities on the attributes, multiplying these utilities by the normalized weights of the attributes, and adding the products. This weighted average utility provides not only an overall evaluation of the system or design, but an appropriate and defensible statement about the seriousness of each deficiency, thus providing rationale for suggested system improvements. Because the utilities and weights are expressed on appropriate ratio scales of value instead of on vague, abstract, or arbitrary scales, the numerical evaluations of an MAU analysis can be legitimately interpreted as comparative assessments and can be used to support decisions.

In a hierarchical MAU analysis, evaluations are “rolled-up” to provide intermediate summary measures as well as an overall measure. These rolled-up values aid in developing an audit trail of the reasoning behind the evaluations. The audit trail can trace from the overall evaluation through the intermediate evaluations to the qualitative descriptors or metrics used to describe the system’s or design’s performance against the lowest level attributes. This important feature of MAU supports the development of strong evaluation or design rationale.

2.4 Sensitivity Analyses

A sensitivity analysis is the last step in an MAU analysis. The results of the analysis may be sensitive to either utilities or weights assigned in the MAU. Four types of sensitivity analyses are most important, and they should be included to run automatically in any MAU decision support tool. First, the utilities that had been assessed could be modified to determine if results change. We have found that results are usually insensitive to minor changes in utilities. Next, several weights could be changed and the overall utility recalculated. This is useful in examining large-scale changes to the model (e.g., for representing a different decision maker's weights), but does not make it easy to isolate causes of change. A third sensitivity analysis is to vary one weight at a time and identify regions where the rankings change. This can isolate the most sensitive inputs. Fourth, the evaluation can be compared against goals and requirements, to identify areas of shortcoming and superior performance. When combined with weights, this type of analysis identifies and isolates the most important areas for improvement. Sensitivity analysis is an important step in an MAU analysis. It helps solidify assessments and identify critical areas for further study. A sensitivity analysis also contributes to the quality of the rationale supporting decisions based on the analysis.

A major strength of MAU analysis is the quality of its measurement techniques that allows strong statements about values. Results of the analysis will include ways to compare, quantify, provide standards, and generate figures of merit (of groups of attributes and metrics and overall). The MAU analysis also supports an audit trail of inputs to outputs and a prioritization of system deficiencies. The method includes a way to trace the result to provide a deeper explanation for the overall figure of merit and to retain the information in qualitative attributes. The method will support decisions based upon metrics. The method will provide a quantification and comparison of attributes over time, the comparison between similar systems, the comparison to requirements, and a measure of utility for a system in a particular environment.

2.5 Extensions to the Basic Method

Two types of interdependence might be important to the uses of MAU described above: 1) interdependence in the *values* of attributes and 2) interdependence in the *technical performance* of alternatives against attributes. These two types of dependencies require different analytical methods. The steps described above are correct if the attributes are additive independent. Loosely speaking, attributes are additive independent if the preferences for levels of an attribute do not depend on the levels of other attributes. Research in decision analysis has produced the theoretical basis for a number of other multiattribute utility formulations. These include: multiplicative, multilinear, generalized multilinear, bilaterally independent, and joint interpolation independent. Each of these special forms, as well as the completely general case, has its own input requirements that are more extensive than those for the additive independent form. In particular, it is impossible to separate the elicitation into separate considerations of the utility functions for individual attributes and weights across attributes. With these non-additive forms, at least some features of multiple attributes must be considered, and elicited, simultaneously. Furthermore, each form has more complicated processing algorithms and more difficult display problems. Nevertheless, these forms offer powerful processing capabilities in cases where value, how much one cares about the attributes, is not additive. This could be the case if how much one cares about an attribute depends on the level of performance on other attributes.

(Note, however, that an additive MAU function is often an excellent approximation to other forms. Edwards (1977, p. 250) states: "Theory, simulation computations, and experience all suggest that weighted linear averages [additive MAU functions] yield extremely close approximations to very much more complicated nonlinear and interactive 'true' utility functions, while remaining far easier to elicit and understand." Edwards cites several studies to back his assertion.)

A particular case, multiplicative, is the simplest extension and one that solves the most serious cases of preference interdependence. It is sometimes the case that poor performance on one attribute is so important that it drives the utility of the system to zero regardless of its performance on other attributes. The additive MAU function cannot address this. All attributes in an additive model are compensatory; poor performance on one attribute can be compensated for by good performance on a different attribute. This situation can be addressed by multiplicative utility functions. If the overall utility is the product of two or more utilities, then if any one is equal to zero, the product is zero. Introduction of some multiplicative groups of attributes into an otherwise additive structure offers a greater richness of possibilities. Here, those highly interdependent attributes can be combined multiplicatively and their product can be traded-off additively against other attributes. Sometimes, a seemingly hopeless number of interdependencies can be represented adequately by a re-structuring of attributes into groups that are multiplicative within each group and additive across groups. (This was explored but found unnecessary in the final MAU structure for UltraLog.)

Sometimes, a better way to address nonadditivity is through the use of thresholds on individual attributes or metrics (or groups of attributes or metrics). A threshold is a value that must be achieved in order for a system to be acceptable. If any threshold is missed, then the system is unacceptable, regardless of its performance on the other attributes. As explained above, this could also be modeled as a multiplicative utility function.

At an extreme, thresholds can be regarded as strict requirements on the system designs—if a design fails, however closely, to meet a threshold, it is worthless. The thresholds thus serve to screen out designs. It is not worth considering any design that does not meet or exceed every threshold. Assessments that fall below threshold level on any attribute could be highlighted as such in the analysis. The highlighting alerts the user to the failure to meet a requirement but the evaluation allows the user to consider other performance characteristics. In the final analysis, those thresholds can be reviewed, and separate decisions can be made on whether the failures are important enough to eliminate alternatives from consideration. Highlights could be carried at all levels in a hierarchical analysis to allow the user to trace the origins of any failure. The use of highlights is illustrated in other sections of this report.

There are at least two situations where this interpretation of thresholds can cause a problem in the analysis. First, thresholds may be set so strictly that there are no feasible system designs. Furthermore, this condition may not be known until a considerable amount of effort is wasted trying to come up with a feasible design. Second, thresholds may not have been meant to be interpreted so strictly. “Requirements” are often meant to be no more than performance goals—important performance levels to try to achieve, but not fatal if missed slightly, especially if performance is close to requirements in some areas and greatly exceeds requirements in other areas. In the extreme, if every attribute had a threshold, then the analysis would reduce to a simple checklist. Such a simple list would also give a simple conclusion, pass or fail. It would not provide a meaningful figure of merit.

The other type of interdependence is technical interdependence among attributes. This is more common. This might be manifested in the appearance of the same measurable quantity in more than one metric. A common example from software engineering is that the attribute, size, might be expressed in the metric, source lines of code, and the attribute, quality, might be expressed in the metric, defects per thousand source lines of code. Clearly these two attributes will be related through the common metric, source lines of code. However, that is of no concern to the MAU analysis, which only needs *preferential* independence of the attributes, i.e., that size and quality are preferentially independent to the decision maker. All of the methods described here work with technically interdependent attributes and metrics.

3. ULTRALOG ASSESSMENT

Multiattribute utility (MAU) analysis was used as a a unifying methodology for UltraLog assessments. MAU was used to make quantitative functional assessments, security assessments, robustness assessments, and scalability assessments. It was used to combine a security assessment, robustness assessment, and scalability assessment into a survivability assessment. It was also used to combine a functional assessment, survivability assessment, and cost assessment into an overall assessment. MAU provided a single, overall figure of merit for an UltraLog system. It also provided figures of merit for individual and groups of functional, security, robustness, and scalability attributes. MAU provided a practical method for making measurable assessments of an UltraLog system against its goals. MAU provided UltraLog's Program management with a strong tool to evaluate and monitor the progress of UltraLog's design and development. It also provided the program with the basis for making a strong case for the eventual pick-up of UltraLog by a service for further development and eventual deployment. Figure 3-1 shows the MAU role in the overall assessment process.

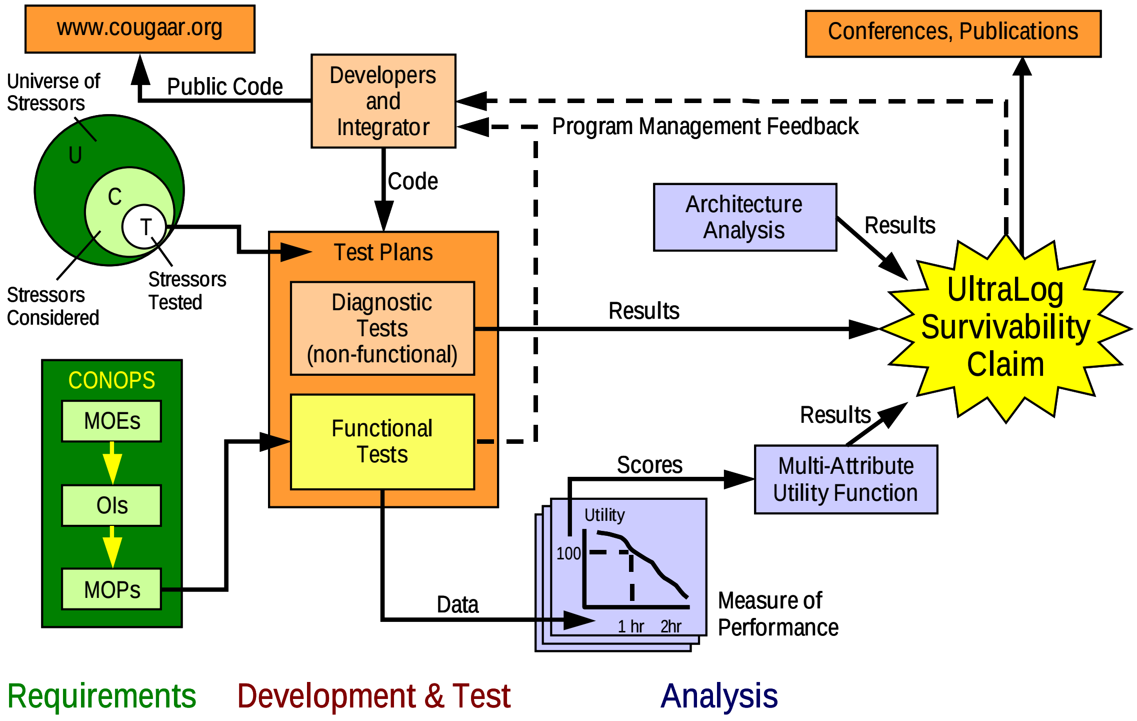


Figure 3-1: MAU in the UltraLog Program Cycle

Figure 3-2 shows the top level of an initial, illustrative but *comprehensive*, hierarchical structure of UltraLog attributes. These are the attributes that a service sponsor will look for in deciding which version, if any, of UltraLog to develop and deploy. It shows that the main attributes of interest are UltraLog's functional performance, its cost, and its survivability. Survivability is here shown to consist of security, robustness, and scalability. (The final version differed from that discussed here for illustration. See Table 6-2.) Figure 3-2 also shows that each of these attributes consists of subattributes (indicated by the branches below each node). The ultimate customer for UltraLog wants a system that performs all of the functions required by military logistics. He wants the system to maintain the security of the logistics plan and to resist or deter attacks on the system's security. He wants the system's performance to be robust; to continue to provide its capabilities and functions at a reasonable level even as it is subjected to kinetic or information warfare attacks. He wants UltraLog to do all of these things at a reasonable cost.

This structure of the assessment as an MAU analysis recognizes that any given UltraLog design or implementation will necessarily make tradeoffs among these attributes. Features added to increase security might reduce functional performance. For example, some security features might slow the system's response to queries. Features that enhance robustness might increase costs. For example, adding redundancy is one way to increase robustness, but redundancy adds more cost by requiring more processors. The MAU analysis provides a framework in which to specify the range of acceptable performance against each desirable feature of the system (attribute). It also provides a means of stating precisely the tradeoffs among attributes. An MAU analysis provides evaluations that reflect the decision maker's or organization's preferences for tradeoffs among all important attributes.

An MAU analysis provides results at any and all nodes in the attribute hierarchy. For Figure 3-2, an MAU analysis provides results as an overall utility that considers all attributes. It also provides results as utilities for the functional assessment, cost, and survivability. It also provides results as utilities for security, robustness, and scalability. Results are also provided for more detailed attributes and groups of attributes that are not shown in the figure. By providing results at all of these levels, the MAU analysis also provides a trace or "audit trail" that helps to explain the reasons for any given evaluation.

MAU analyses for groups of attributes might also be of interest in their own right, especially in the design and development of UltraLog. The MAU analysis of functional attributes could be of interest to improve functional performance. The MAU analysis of security could highlight current security weaknesses and indicate which weakness are most important to improve. The MAU analysis of robustness could be used to improve the robustness of UltraLog to kinetic and information warfare attacks. Repeated MAU analyses over time with later versions of UltraLog can demonstrate improvement and progress toward goals.

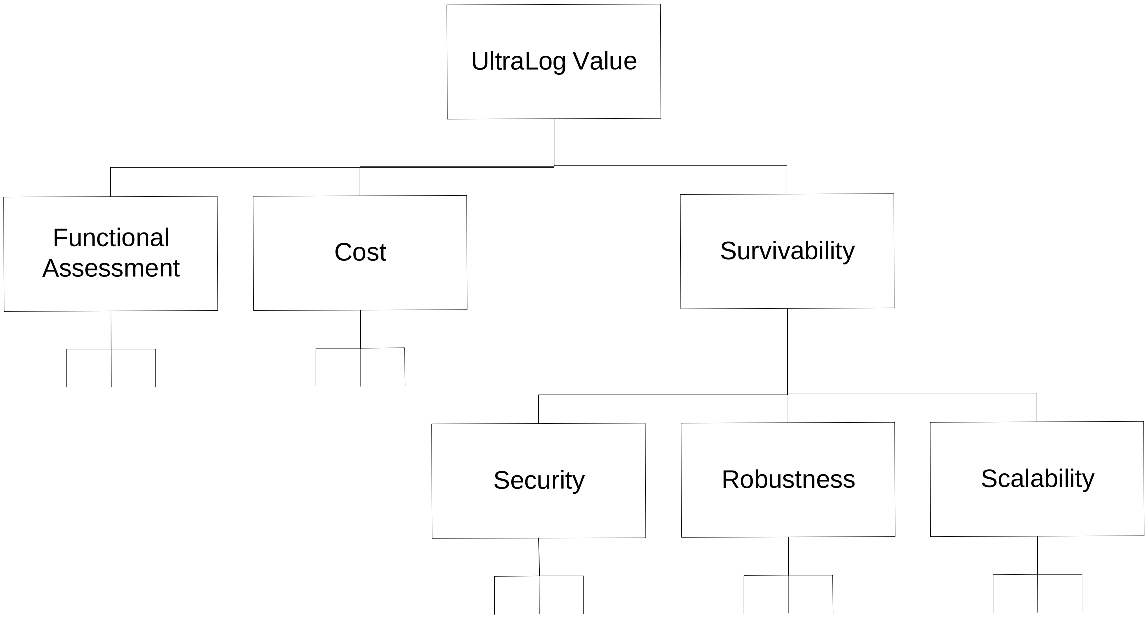


Figure 3-2. Hierarchy of attributes for an MAU analysis of UltraLog.

Specific illustrations of MAU analyses were developed prior to eliciting and assessing the structure and utility curves actually employed later. These *guiding* illustrations are presented in the following sections to illustrate how the work was done. Section 4 contains an illustrative MAU functional assessment. Section 5 contains an MAU hierarchy of security attributes and an illustrative MAU Red Team security assessment. Sections 6 and 7 contain background analyses that feed into the illustrative MAU robustness assessment in Section 8. No cost analysis or scalability analysis has been conducted for UltraLog.

MAU analysis as described in this report provided a unifying methodology to combine the efforts of the UltraLog contractors. For example, a complete specification of an MAU analysis provided overall direction to the contractors by showing: quantitative goals for performance; tradeoffs within and among attributes of security, functionality, and robustness; and the relative importance of deficiencies discovered during experiments and tests.

The MAU structure itself evolved during the work. Various forms were tested and experts were used to determine weaknesses and improvements. The following Sections 4 and 5 present the illustrative approaches and initial MAU models that were developed and discuss the illustrative examples and results that were developed to guide the refinement of the models, utility functions, and weights. Section 6 shows how the model structure was further revised and developed and utility functions and weights were assessed during a series of structured conferences with experts.

4. FUNCTIONAL ASSESSMENT

This section describes how multiattribute utility (MAU) analysis methods were used as part of a functional assessment of UltraLog. This section follows the six steps of an MAU analysis described in Section 2: structure attributes, specify utility curves, establish weights across attributes, assess performance of systems against attributes, calculate results, and conduct sensitivity analyses. It also provides a complete, illustrative MAU example that was used as a starting point for actual elicitations, such as those described in Section 6.

All numbers contained in this section (as well as other section of this paper) are hypothetical and are for purposes of illustration only. However, the metrics and their Goal and Minimum (worst) levels were developed during meetings with John Benton and Leo Pigaty of Los Alamos Technical Associates, Inc. Utility functions were developed with and reviewed by John Benton.

4.1 Step 1: Structure Attributes for an UltraLog Functional Assessment

The first step in an MAU analysis is to develop a structure of the attributes. The functional aspects of UltraLog are described in the “Ultra Log Functional Assessment (7/17/01),” which was developed by Los Alamos Technical Associates, Inc. That document details the functional requirements for UltraLog by describing two measures of effectiveness (MOEs), six operational impacts (OIs), and thirty-two measures of performance (MOPs). It is reproduced as Appendix A to this report.

The organization of the functional assessment lends itself naturally to a multiattribute utility (MAU) hierarchy of UltraLog functional attributes. The two MOEs describe completely the functional attributes of UltraLog. Each MOE is described completely by three OIs, and each OI is described completely by a set of three to ten MOPs.

Figure 4-1 displays an MAU hierarchy based on the functional assessment. The top level is the UltraLog functional assessment. The functional assessment consists of two MOEs, Provide Warfighting Information, and Operate Effectively. Provide Warfighting Information consists of three OIs: produce an executable logplan, provide timely information, and be adaptive. Each of these OIs comprises three to ten MOPs, as shown. The second MOE, Operate Effectively, consists of three OIs: interoperate with existing communications, remain mission ready and operating, and remain user friendly. Each of these OIs comprises from three to six MOPs.

As structured, the set of MOPs under each OI is meant to capture completely the OI. The number of MOPs used to describe the OIs varies from three to ten. More MOPs under an OI does not imply that the OI is more important, just that it requires more detail to describe the OI. (The numbers shown in Figure 4-1 are described in Section 4.3).

4.2 Step 2: Specify a Utility Function for Each Functional Attribute

The second step in an MAU analysis is to specify a utility function for each attribute. The utility function describes how much the decision maker cares about changes in the attribute. Utility curves were developed in two steps. First, a metric was described for each attribute and values corresponding to the goal and worst acceptable levels were identified. Then, curves were assessed for the other values of the metric.

Metrics for the MOPs are given in Appendix A. A mixture of metrics and qualitative descriptors of value were described for the MOPs. Most metrics referred to the time that it would take UltraLog to respond or perform a task, such as answer a query, or to the accuracy with which UltraLog would provide information. Qualitative descriptors were usually four-point scales defined by logistics experts.

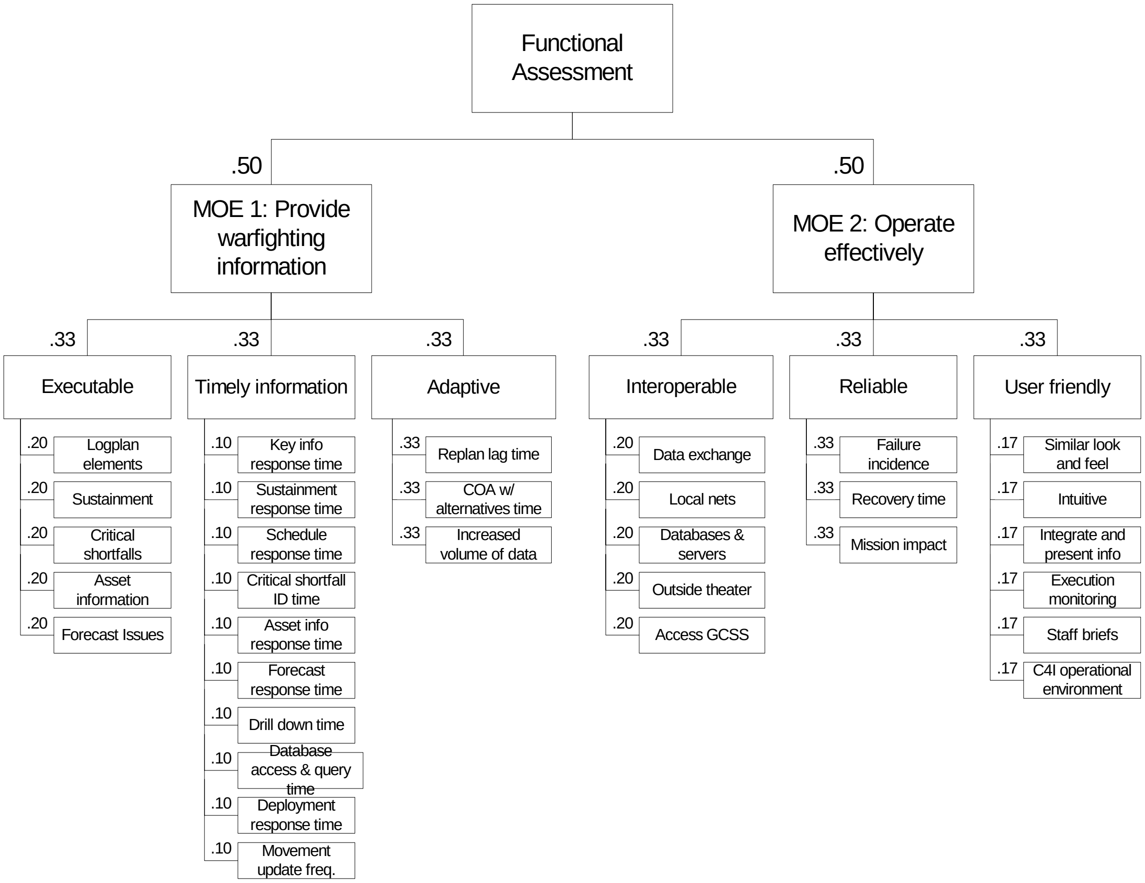


Figure 4-1. Attributes of a functional assessment.

For all utility functions, a utility of 100 was assigned to meeting the goal for performance on the attribute. For example, the goal of an average error of 5% in the estimation of key logplan elements one day out was assigned a utility of 100. The goal of responding to 95% of the queries about key logplan element within 30 seconds was assigned a utility of 100. The goal of producing level six deployment data for a contingency within 1 hour was assigned a utility of 100. Consistently assigning a utility of 100 to the goal level provides a reference utility for all places in the analysis; whenever a system has a utility of 100, it is as valuable as a system that meets all goals fully.

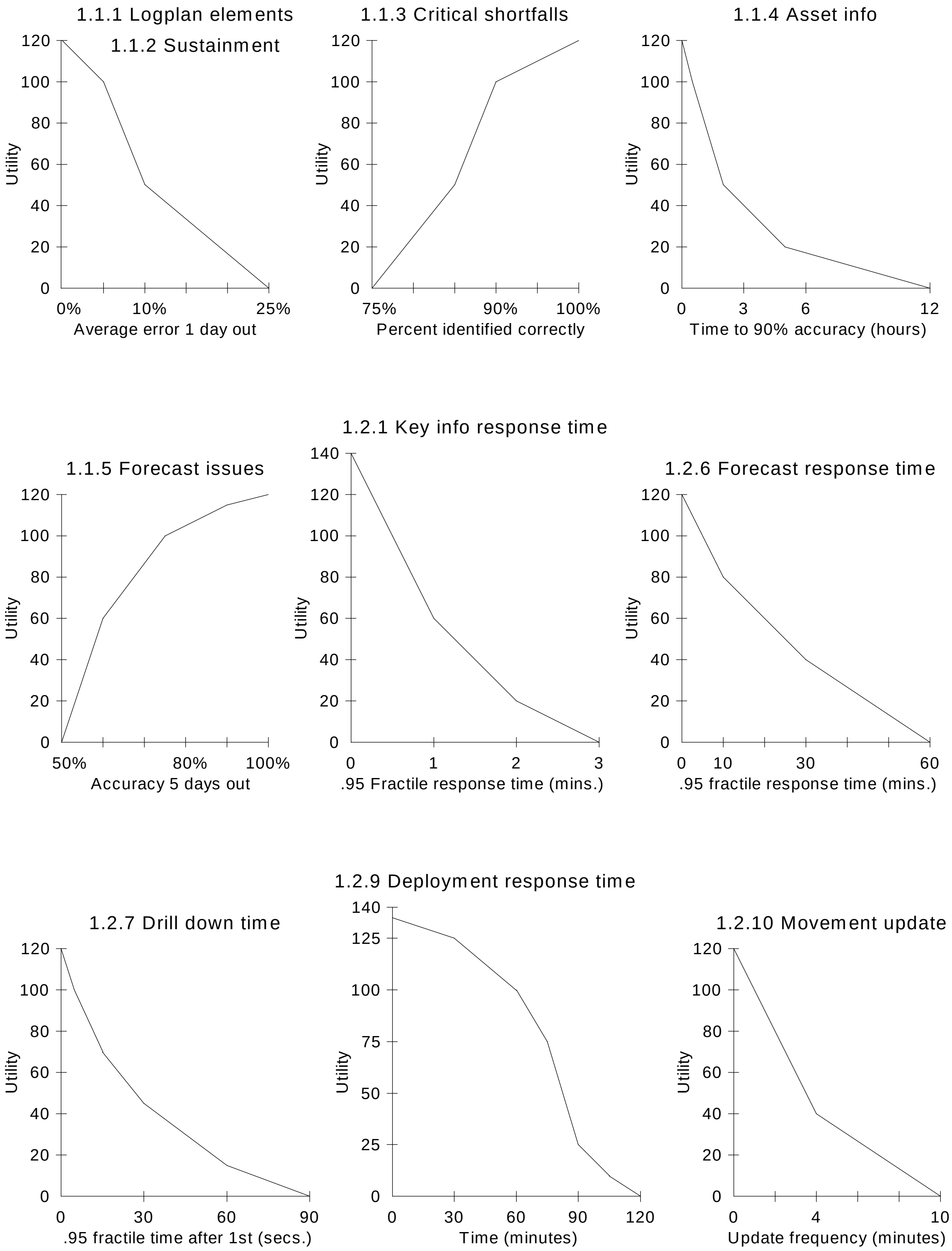
Similarly, the worst acceptable level of performance on an attribute was assigned a utility of 0. For example, an average error of 25% in the estimation of key logplan elements one day out was assigned a utility of 0. Responding to 95% of the queries about key logplan element within 3 minutes was assigned a utility of 0. Producing level six deployment data for a contingency within 2 hours was assigned a utility of 0. Consistently assigning a utility of 0 to the worst acceptable level provides a reference value for highlighting unacceptable performance. Any system that does not provide at least the worst acceptable level on all attributes is unacceptable.

Since utility is defined uniquely only up to a linear transform, assigning the 0 and 100 points on the utility function is perfectly acceptable. Assigning a utility of 100 to the goal level of performance opens the possibility that a system may perform better than the goal. For example, key logplan elements one day out might be estimated with an average error of less than 5%, possibly with no error. 95% of the queries about key logplan elements could be made instantaneously. Level six deployment data for a contingency could be provided in much less than 1 hour. In these cases, utilities of greater than 100 could be assessed, if that improved level of performance is preferred over meeting the goal. The amount of increase is the value relative to the value of improving from the worst acceptable level to the goal. If the best possible is 25% of that range better than the goal, then the best possible level is assigned a utility of 125. Utility of 100 is the goal, not the maximum.

Figure 4-2 shows the utility functions assessed for the functional attributes. Notice that the curves exhibit a variety of shapes: linear, approximately exponential, and S-shaped, increasing and decreasing. (All curves are piece-wise linear between assessed points. Piece-wise linear curves can approximate any shape with a degree of error that depends on the number of pieces.) The utility for the time that the system takes to respond to 95% of the queries about key information (attribute 1.2.1) drops off approximately exponentially from instantaneous (with a utility of 140), through the goal of 30 seconds, to the worst acceptable time of 3 minutes. The utility for deployment response time (attribute 1.2.9) is a reflected S-shape. Utility drops quickly as this time increases above the goal of 60 minutes to 90 minutes. Utility drops slowly between 90 minutes and the worst acceptable time of 120 minutes. Utility rises slowly for times less than 60 minutes up to a maximum utility of 135. (It would have been possible to specify a utility function that did not rise, or even fell, for improvements over a given value.) A number of curves are the same. For example the ability to interoperate with local networks (attribute 2.1.2) and with databases and servers (attribute 2.1.3) have linear utility curves for the range of 95% to 100% interoperability.

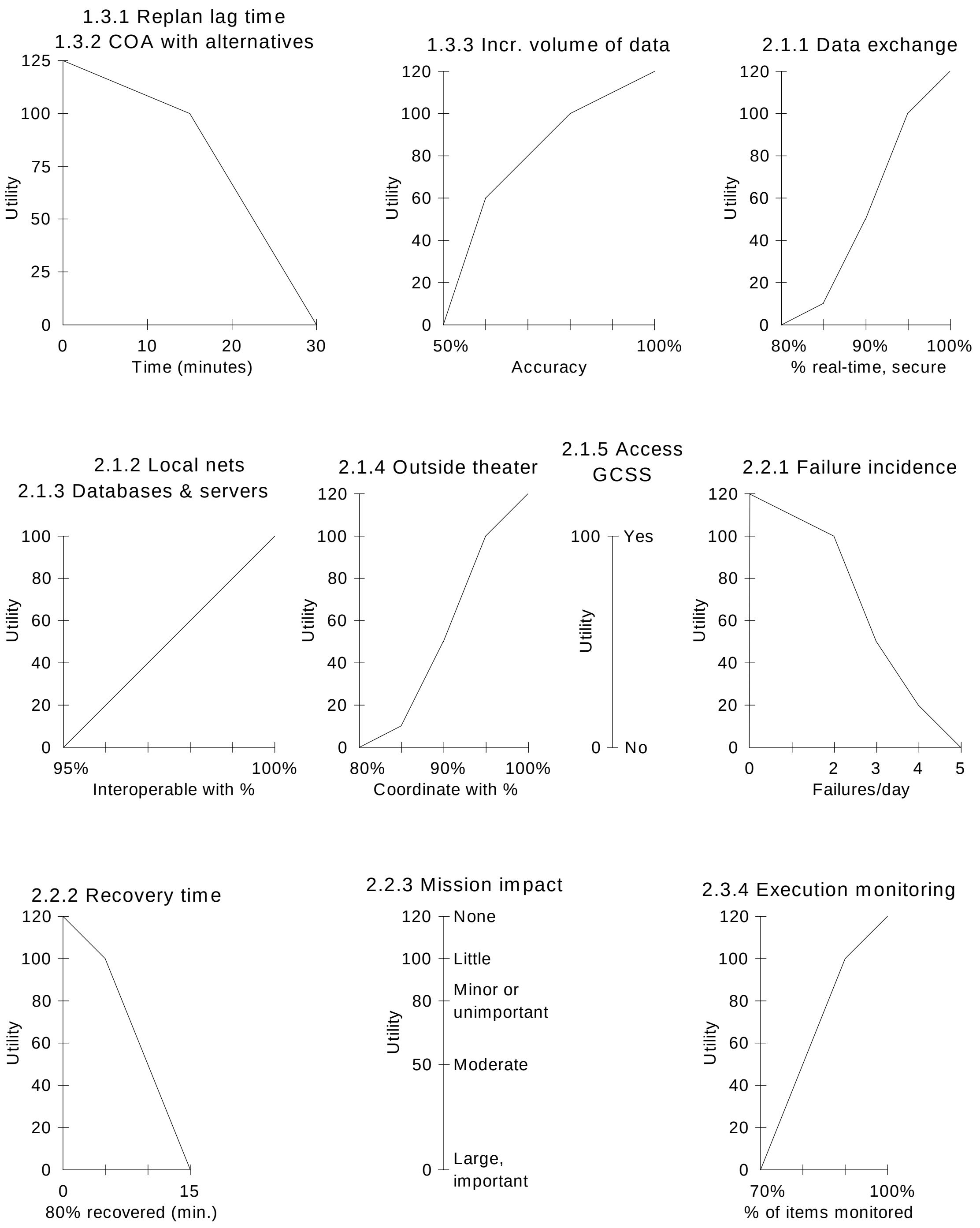
Notice that some of the utility functions have been simplified for this presentation. In particular, functional assessment metrics that had more than one quantity were simplified to a single quantity. For example, the metric for MOP 1.1.1, Logplan elements, refers to the average error one day out and five days out. The utility function shows utility over the one day out error. A refinement of this analysis could treat both aspects of these utility functions.

Also notice that a number of utility functions refer to the 0.95 fractile. Performance against these metrics is expected to be uncertain. These uncertainties could be represented by probability distributions. The 0.95 fractile of the distribution is the value of the metric (usually time) that is to be met 95% of the time. For example, if 95% of queries are met within 30 seconds, then the 0.95 fractile of the distribution of



1.2.2 Sustainment response time, 1.2.3 Schedule response time, 1.2.4 Critical shortfall ID time, 1.2.5 Asset information response time, and 1.2.8 Database access & query time all have the same utility curve as 1.2.1 Key info response time.

Figure 4-2. Utility functions for functional assessment attributes.



2.3.1 Similar look and feel, 2.3.2 Intuitive, 2.3.3 Integrate and present info, 2.3.5 Staff Briefs, and 2.3.6 C4I operational environment all have qualitative scales similar in form to the one shown for 2.2.3 Mission impact.

Figure 4-2. Utility functions for functional assessment attributes (continued).

query times is at 30 seconds. If 95% of queries are met within 3 minutes, then the 0.95 fractile of the distribution of query time is at 3 minutes.

4.3 Step 3: Establish Weights Across MOEs, OIs, and MOPs

The third step in an MAU analysis is to establish tradeoffs across attributes. In an additive MAU analysis, tradeoffs are expressed as weights. The weights are used to compare utilities across attributes and to combine the utilities on single attributes into utilities on groups of attributes. When assessing weights, one compares the importance of the swing on one attribute scale with the swings on other attribute scales.

Figure 4-1 shows a set of weights in the attribute hierarchy. In this case, weights were set equal within each level of the hierarchy. This reflected the following reasoning. At the top level, functionality consists of two measures of effectiveness, providing warfighting information and operating effectively. These measures of effectiveness are equally important. Within the first MOE, the three operational impacts, executable, timely information, and adaptive, are equally important. Each operational impact is characterized by a different number of measures of performance (MOPs). However, each set of MOPs describes fully its corresponding operational impact. Within each operational impact, the range of performance from the minimum acceptable to the goal is equally important for each MOP. The same is true for the operational impacts and MOPs under the second measure of effectiveness.

The assessed weights are normalized to add to 1.0 at each level in the hierarchy of attributes. These normalized weights are shown in Figure 4-1.

Table 4-1. Functional assessments of UltraLog performance (for illustration only).

Attributes	UltraLog 01	UltraLog 02	UltraLog 03	Goal
1.1.1 Logplan elements	10%	7%	5%	5%
1.1.2 Sustainment (1 day)	30%	10%	6%	5%
1.1.3 Critical shortfalls	86%	90%	92%	90%
1.1.4 Asset info (90%)	5 hours	2 hours	0.75 hour	0.5 hour
1.1.5 Forecast issues (5 day)	55%	70%	85%	75%
1.2.1 Key info response time	2 mins.	1 min.	0.5 min.	0.5 min.
1.2.2 Sustainment res. time	2.5 mins.	1.5 mins.	0.75 min.	0.5 min.
1.2.3 Schedule response time	1.5 mins.	1 min.	0.5 min.	0.5 min.
1.2.4 Critical shortfall ID time	1 min.	0.75 min.	0.5 min.	0.5 min.
1.2.5 Asset info response time	2 mins.	1 min.	0.5 min.	0.5 min.
1.2.6 Forecast response time	45 mins.	30 mins.	10 mins.	5 mins.
1.2.7 Drill down time	45 secs.	30 secs.	10 secs.	5 secs.
1.2.8 Database access & query	2 mins.	1.5 mins.	0.5 min.	0.5 min.
1.2.9 Deploym'nt response time	100 mins.	80 mins.	65 mins.	60 mins.
1.2.10 Movement update	8 mins.	4 mins.	1.5 mins.	1 min.
1.3.1 Replan lag time	28 mins.	24 mins.	16 mins.	60 mins.
1.3.2 COA with alternatives	40 mins.	22 mins.	15 mins.	15 mins.
1.3.3 Increased data volume	40%	55%	70%	80%
2.1.1 Data exchange	75%	90%	93%	95%
2.1.2 Local nets	96%	98%	100%	100%
2.1.3 Databases and servers	95%	97%	99%	100%
2.1.4 Outside theater	85%	90%	95%	95%
2.2.1 Failure incidence	4.5 fails/day	3.5 fails/day	2.5 fails/day	2 fails/day
2.2.2 Recovery time (80%)	15 mins.	10 mins.	7 mins.	5 mins.
2.3.4 Execution monitoring	75%	80%	90%	90%

4.4 Step 4: Assess Performance of Systems Against Attributes

The next step in an MAU analysis is to specify the performance of the system under evaluation against each of the attributes. For purposes of illustration, assume that the UltraLog system is evaluated each year, 2001, 2002, and 2003 (labeled UltraLog 01, 02, and 03, respectively). In this step of the analysis, an assessment is made of the performance of the system against each measure of performance. An example of the results of such an assessment is shown in Table 4-1. This table also shows assessments for a hypothetical system that meets the goal on every attribute.

These performance assessments are then transformed into utility assessments using the utility functions presented in Section 4.2. (For MOPs with qualitative descriptors, utilities were assigned to the descriptors.) This results in the utility assessments shown in Table 4-2. Several places show performance worse than the worst acceptable. These performances are designated (0) in the utility table. This serves as a highlight that performance against the MOP is unacceptable. In the calculation described in Section 4.5, these will be treated as zero utility against the MOP, but the highlight will be associated with all calculations to alert the user to the condition that performance is unacceptable.

Table 4-2. Utility assessments of UltraLog performance (for illustration only).

Attributes	UltraLog 01	UltraLog 02	UltraLog 03	Goal
1.1.1 Logplan elements	50	80	100	100
1.1.2 Sustainment (1 day)	(0)	50	90	100
1.1.3 Critical shortfalls	50	100	104	100
1.1.4 Asset info (90%)	20	50	92	100
1.1.5 Forecast issues (5 day)	30	87	110	100
1.2.1 Key info response time	20	60	100	100
1.2.2 Sustainment res. time	10	40	80	100
1.2.3 Schedule response time	40	60	100	100
1.2.4 Critical shortfall ID time	60	80	100	100
1.2.5 Asset info response time	20	60	100	100
1.2.6 Forecast response time	20	40	80	100
1.2.7 Drill down time	30	45	85	100
1.2.8 Database access & query	20	40	100	100
1.2.9 Deploym’nt response time	15	58	92	100
1.2.10 Movement update	13	40	90	100
1.3.1 Replan lag time	13	40	93	100
1.3.2 COA with alternatives	(0)	53	100	100
1.3.3 Increased data volume	(0)	30	80	100
2.1.1 Data exchange	(0)	50	80	100
2.1.2 Local nets	20	60	100	100
2.1.3 Databases and servers	0	40	80	100
2.1.4 Outside theater	10	50	100	100
2.1.5 Access GCSS	0	100	100	100
2.2.1 Failure incidence	10	35	75	100
2.2.2 Recovery time (80%)	0	50	80	100
2.2.3 Mission impact	10	60	90	100
2.3.1 Similar look and feel	100	100	100	100
2.3.2 Intuitive	100	100	100	100
2.3.3 Integrate and present info	90	100	100	100
2.3.4 Execution monitoring	25	50	100	100
2.3.5 Staff briefs	80	100	100	100
2.3.6 C4I operational environ.	60	100	100	100

4.5 Step 5: Calculate the Functional Assessment

The fifth step in an MAU analysis is to calculate the utility of the systems at all levels in the hierarchy of attributes. Starting at the bottom level, Table 4-3 shows the calculation of the “rolled up” utility against the first operational impact, 1.1 Executable.

Table 4-3. Calculation of utility for 1.1 Executable.

	weight	UltraLog 01	UltraLog 02	UltraLog 03	Goal
1.1 Executable		**(30)**	**73**	**99**	**100**
1.1.1 Logplan elements	0.20	50	80	100	100
1.1.2 Sustainment	0.20	(0)	50	90	100
1.1.3 Critical shortfalls	0.20	50	100	104	100
1.1.4 Asset information	0.20	20	50	92	100
1.1.5 Forecast issues	0.20	30	87	110	100

In an additive MAU analysis, alternatives are evaluated by calculating a weighted multiattribute utility. The weights are the normalized swing weights discussed in Section 4.3. For example, the evaluation of UltraLog 02 against the operational impact, Executable, is calculated as follows:

$$(0.20)(80) + (0.20)(50) + (0.20)(100) + (0.20)(50) + (0.20)(87) = 73.$$

(All calculations are rounded for display to two decimal places for swing weights and to whole numbers for utilities.) The calculated utility for UltraLog 01 is shown in parentheses to highlight the fact that UltraLog 01 failed to achieve the minimally acceptable level of performance on an MOP. Looking down the column, one sees that the unacceptable performance was against MOP 1.1.2 Sustainment. The highlight can be traced back to the assessed performance in Table 4-1 to see that UltraLog 01 could estimate gross resupply and other sustainment requirements one day out with an average error of 30%. However, as shown in Figure 4-2, the worst acceptable performance on this attribute was assessed at an error of 25%. (Recall that the utility function for each attribute assigned a utility of 0 to the worst acceptable level.) By performing the calculation in this manner while highlighting the deficiency, one is able to: highlight the fact that the system performed unsatisfactorily on an attribute, identify the attribute where performance was unsatisfactory, and show what the evaluation of the system would have been had it performed at the minimal acceptable level on that attribute (i.e., 30 on the OI, Executable).

Table 4-4. Calculation of functional assessment utilities.

	weight	UltraLog 01	UltraLog 02	UltraLog 03	Goal
FUNCTIONAL ASSESSMENT		**(25)**	**61**	**93**	**100**
1 MOE 1: Warfighting Info	.50	(20)	56	94	100
1.1 Executable	.33	(30)	73	99	100
1.2 Timely information	.33	25	52	93	100
1.3 Adaptive	.33	(4)	41	91	100
2 MOE 2: Operate effectively	.50	(30)	67	91	100
2.1 Interoperable	.33	(6)	60	92	100
2.2 Reliable	.33	7	48	82	100
2.3 User friendly	.33	76	92	100	100

Similar calculations are repeated for all of the operational impact attributes. The MAU evaluation against each MOE is then calculated as a weighted average of the utilities against the operational impacts. The functional assessment utilities are calculated as weighted averages of the MOE utilities. These calculations are shown in Table 4-4.

The following paragraphs explain how these results can be used by system developers and evaluators.

What is the functional assessment of the current UltraLog development? The multiattribute utility functional assessments are shown in the top row of Table 4-4. The performance of a particular version of UltraLog can be evaluated as described above. The table shows, for example, that UltraLog 01 has an evaluation of (25). This evaluation is interpretable as a functionality that is 25% of the way toward the goal. However, the parentheses indicate that UltraLog 01 fails to meet the minimum acceptable performance on at least one MOP, so it is unacceptable. The utility functions were constructed to be cardinal scales so the evaluation numbers carry more information than a simple ordering, and this information is useful for answering the next questions.

How does this compare with the requirements and goals? The goal represents a hypothetical system that contains the goal level of functionality on every MOP. It has a utility of 100 for each MOP, and its overall utility is 100. The MAU evaluations of the systems can be compared with the goal. Table 4-4 shows, for example, that UltraLog 03 meets the goal for one OI, User Friendly, but falls short on all of the others. Its overall evaluation of 93 indicates that it is 93% of the way from a minimally acceptable system to the goal.

How does this compare with the baseline? Suppose that UltraLog 01 is taken as the baseline. Table 4-4 shows that UltraLog 01 has a functional assessment utility of 25. However it also shows that UltraLog 01 performs below the minimal acceptable level on two operational impacts. Thus, UltraLog 02 offers an improvement over the baseline by both meeting every minimal requirement and by improving to 61% of the goal. UltraLog 03 offers an improvement over the baseline by both meeting every minimal requirement and by improving to 93% of the goal.

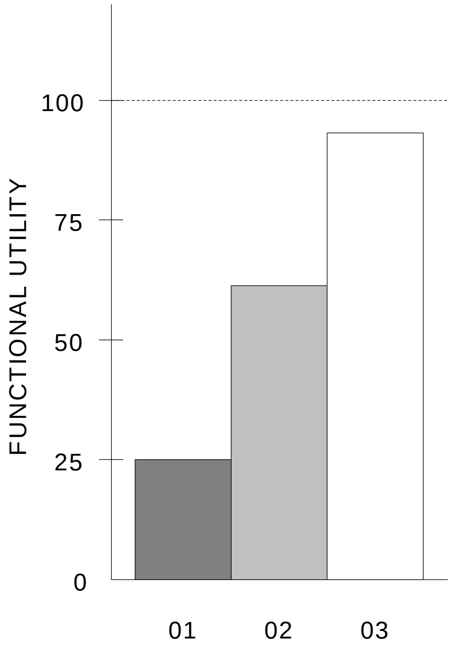


Figure 4-3. Functional assessments of UltraLog over time.

How are the evaluations changing over time? If UltraLog 01, UltraLog 02, and UltraLog 03 are subsequent versions of UltraLog, then an examination of their evaluations shows how evaluations are changing over time. Figure 4-3 shows UltraLog 01 at 25% of the goal, UltraLog 02 improves to 61% of the goal, and UltraLog 03 improves to 93% of the goal. Figure 4-4 shows improvements over time for the six operational impacts.

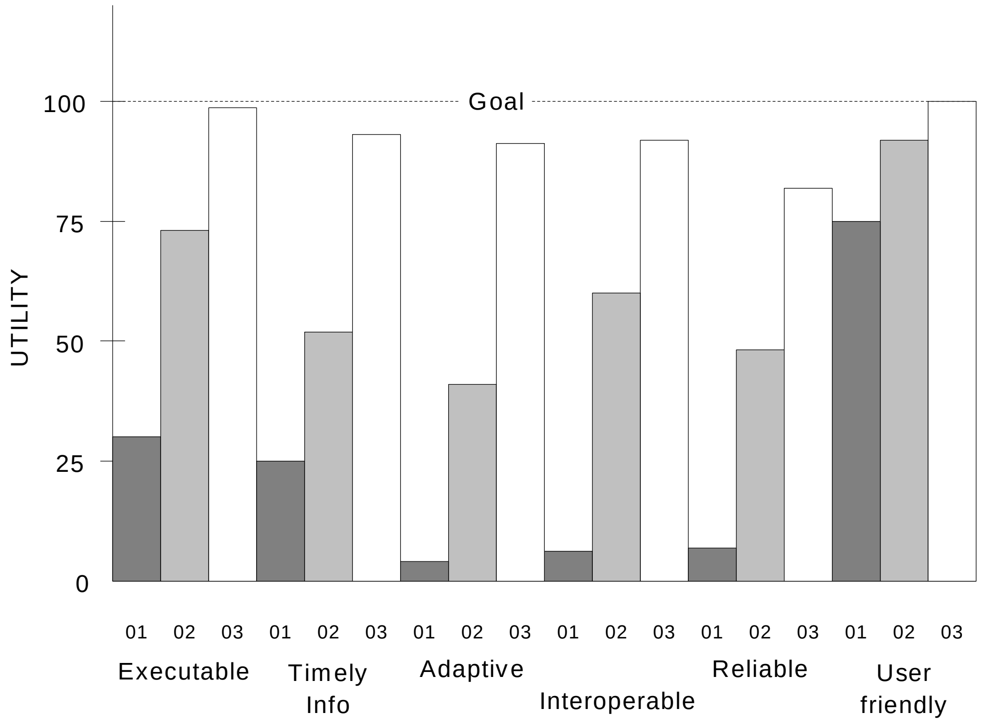


Figure 4-4. Operational impact assessments of UltraLog over time.

4.6 Sensitivity Analyses for the UltraLog Functional Assessment

A sensitivity analysis is an important part of a complete MAU analysis. Sensitivity analyses highlight the sensitive parts of the initial analysis, focus debate on important parts of the analysis, and help to resolve disagreements. Sensitivity analyses can show that more information is necessary or show when more information is unnecessary in order to make a decision or evaluation. For the analysis in the example, a sensitivity analysis could be conducted by changing any of the inputs and examining the resulting changes in evaluations.

A more systematic sensitivity analysis can help to answer the question, *What attributes need to be enhanced to improve the functional assessment?* Any subattribute with a utility of less than the maximum could be improved to improve the functional assessment. There is room for improvement in all of the systems in the example, including the hypothetical Goal system. This question is best answered by looking at a combination of weights and utilities. One feature of the prototype tool described in Section 9 is that it performs this type of analysis automatically. The results are shown in Table 4-5. Here, the performance of UltraLog 02 is compared with the Goal. Table 4-5 shows, in decreasing order of importance, the contributions of the performance on the 32 MOPs to the thirty-nine point difference in utility between UltraLog 02 and the Goal.

This sensitivity analysis shows that UltraLog 02 falls most seriously short of the goal in MOP 1.3.3, Increased volume of data. Performance on this attribute alone accounts for 10% of UltraLog 02's deficiencies (3.9 utility points out of a total difference of 39 points). This is because UltraLog 02 is 70 utility points below the goal on this attribute, and the attribute has a weight in the analysis of

(0.5)(0.333)(0.333) = 0.0555 (which is calculated by multiplying the weights for all branches from the top of Figure 4-1 down to the MOP). The product of the utility difference times this weight is: (70)(0.555) = 39. The top seven attributes contribute over half of the deficiency. The analysis also shows that UltraLog 02 meets the goal on seven of the attributes (those shown at the bottom of the table). If UltraLog 02 had exceeded any of the goals, then those attributes would have been shown at the bottom of the table with negative numbers in the "weighted difference" column. For example, when UltraLog 03 is compared with the Goal, this type of sensitivity analysis shows that the two attributes on which UltraLog is better than the Goal, 1.1.3 Critical shortfalls and 1.1.5 Forecast issues, contribute -0.5 to the comparative evaluation.

Table 4-5. Areas for improving UltraLog 02 compared with the Goal.

Attributes	Weighted difference	Cumulative
1.3.3 Increased volume of data (accuracy)	3.89	3.9
2.2.1 Failure incidence	3.61	7.5
1.3.1 Replan lag time	3.33	10.8
2.2.2 Recovery time (80%)	2.78	13.6
1.3.2 COA with alternatives time	2.59	16.2
2.2.3 Mission impact	2.22	18.4
2.1.3 Databases and servers	2.00	20.4
1.1.2 Sustainment (1 day)	1.67	22.1
1.1.4 Asset information (90%)	1.67	23.8
2.1.1 Data exchange	1.67	25.4
2.1.4 Outside theater	1.67	27.1
2.3.4 Execution monitoring	1.39	28.5
2.1.2 Local nets	1.33	29.8
1.2.2 Sustainment response time	1.00	30.8
1.2.6 Forecast response time	1.00	31.8
1.2.8 Database access & query time	1.00	32.8
1.2.10 Movement update frequency	1.00	33.8
1.2.7 Drill down time	0.92	34.7
1.2.9 Deployment response time	0.69	35.4
1.1.1 Logplan elements (1 day)	0.67	36.1
1.2.1 Key information response time	0.67	36.8
1.2.3 Schedule response time	0.67	37.4
1.2.5 Asset information response time	0.67	38.1
1.1.5 Forecast issues (5 day)	0.44	38.5
1.2.4 Critical shortfall ID time	0.33	38.9
1.1.3 Critical shortfalls	0.00	38.9
2.1.5 Access GCSS	0.00	38.9
2.3.1 Similar look and feel	0.00	38.9
2.3.2 Intuitive	0.00	38.9
2.3.3 Integrate and present information	0.00	38.9
2.3.5 Staff briefs	0.00	38.9
2.3.6 C4I operational environment	0.00	38.9

This sensitivity analysis can also be used to track the improvements from year to year. For example, an analysis comparing UltraLog 02 with UltraLog 03 shows that almost one-third of the 32-point difference in utility comes from four of the thirty-two attributes.

5. SECURITY

This section describes the initial approach to security attributes. As in the previous Section, these served as a test case and a starting point for further development. The final version is shown in Table 6-2 in Section 6. Descriptions are provided for both Red Team security attributes and other security attributes. An illustrative MAU analysis of Red Team attributes is presented.

Figure 5-1 shows some details of the proposed subdivision of Security. This was to guide both assessments by Red Team attacks on UltraLog as it was developed as well as by other means by other means, and these two possibilities are shown as subdivisions of Security. For the present example, we concentrate on Red Team attributes. Saydjari, Wood, and Bouchard (2001) specified six Red Team flags: 1) delay or prevent planning or replanning beyond one hour, 2) make planning data available to an unauthorized entity, 3) cause UltraLog to base a plan or replan on incorrect data, 4) compromise the security state (status or accountability) of UltraLog, 5) affect negatively a non-UltraLog neighbor system, and 6) obtain unauthorized privileges within UltraLog. (These flags were revised and presented as Red Team System Security Flags by Raymond Parks at the UltraLog Developers Workshop in July 2001. The example in this section illustrates the method using the older flags.)

Figure 5-2 shows a possible subdivision of the Other Security attribute. This particular subdivision was developed by examining the information assurance attributes described by Ulvila et al. (2001a) and selecting those attributes that were regarded as being most appropriate for UltraLog.

The first subdivision of Other Security is into Vulnerabilities of the UltraLog system and Countermeasures within UltraLog that mitigate against possible vulnerabilities. Security of UltraLog could be increased either by reducing Vulnerabilities or by increasing Countermeasures, or by doing both.)

Vulnerabilities are weaknesses in an information system, system security procedures, internal controls, or implementation that could be exploited. Vulnerabilities include: Asset Value and Exposure, Malicious Capabilities, Procedural Errors, and Structural Vulnerabilities. Asset value and Exposure includes the value of UltraLog assets (hardware, software, firmware, and information) that are exposed to the threat, and the extent to which they are exposed. Malicious Capabilities are the adversarial capabilities (e.g., training, resources, tools, and opportunity) required to conduct a potentially successful attack. A system is less vulnerable if a successful adversary must be more motivated, be more capable, and expend more resources. Procedural errors are the extent to which user or administrator errors can degrade security operations. Structural vulnerabilities represent the extent to which security is reduced due to structural errors.

Countermeasures are actions, devices, procedures, techniques, or other measures that reduce the vulnerability of the system. Countermeasures comprise Security Functionality and Countermeasure Effectiveness. Security functionality is the totality of mechanisms that support enforcement of the system’s security policy. Security Functionality is divided into Confidentiality, Integrity, Availability, Accountability, Security Protection, and Security Management features. Confidentiality is the extent to which features prevent unauthorized disclosure of critical assets and resources. Integrity is the extent to which features prevent unauthorized modification of critical assets and resources. Availability is the extent to which features prevent unauthorized denial of service to legitimate users. Accountability is the extent to which features provide a capability to audit events so that the initiator of an action can be identified and held accountable. Security protection is the extent to which features provide the system’s capability to protect its own information and to protect the security services that it provides. Security management is the extent to which the security management features (e.g., configuration, procedures,

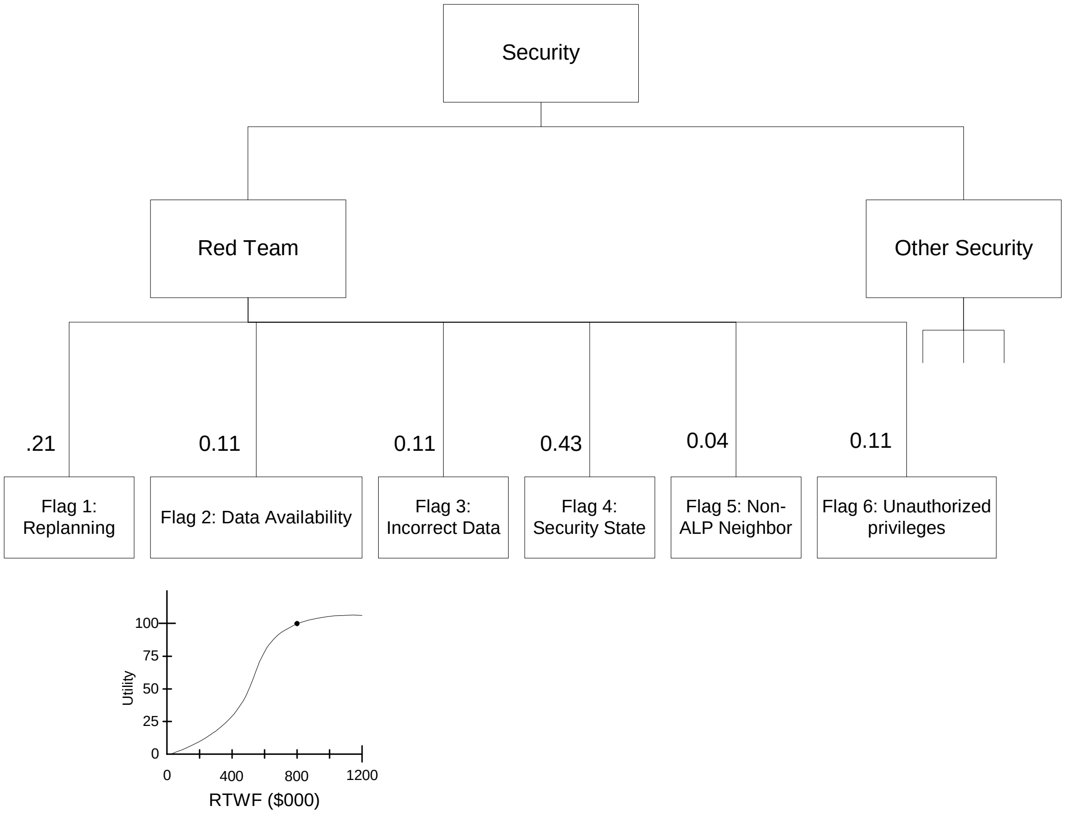


Figure 5-1. Security attributes (details on Red Team attributes).

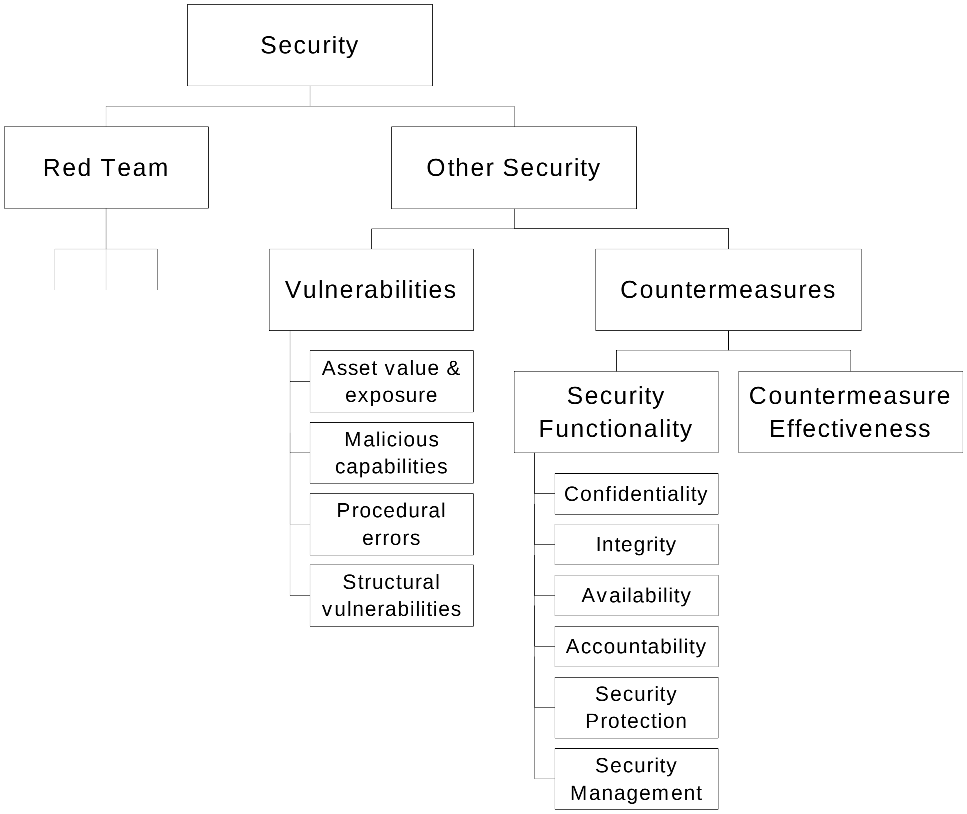


Figure 5-2. Security attributes (details on Other Security attributes).

administration) are automated. Countermeasures effectiveness is the extent of security process compliance, protection, maturity, efficiency, and effectiveness followed in an UltraLog development.

By including both Red Team attributes and Other Security attributes, we recognize that multiple approaches are necessary to ensure the security of UltraLog. Tests and experiments should seek to uncover security problems using both Red Teams and traditional security analysis means (e.g., security design and requirements analysis, security functional analysis and testing, security documentation analysis and review).

5.1 Red Team Utility Functions

The next step is to develop a utility function for each attribute. A utility function describes how much the decision maker cares about changes in the level, degree, or state of attainment of an attribute.

Consider the Red Team attributes first. A better UltraLog (i.e., one with a higher value) will require more effort for a Red team to achieve its flags. The following is an example of the reasoning that might go into specifying a utility function for Flag 2 as shown in Figure 5-1. The particular values used in the example are completely hypothetical. Suppose that the goal is to require $800,000 of RTWF for the Red Team to make planning data available to an unauthorized entity (Flag 2).(Saydjari, Wood, and Bouchard, 2001, describe RTWF and how a variety of its components, e.g., skill and effort of the adversary and attack development tools, can be reduced to a common unit of dollars.) Suppose further that it is completely unacceptable for the Red Team to achieve Flag 2 with an expenditure of less than $20,000. That is, if the Red Team took less than $20,000 to achieve this flag, then the evaluation against Red Team Security would be that UltraLog is “worthless” regardless of how well it performed on the other flags. The graph in Figure 5-1 shows an S-shaped curve for the utility over the amount of RTWF that is required to reach flag 2. Utility rises slowly as the amount of effort increases above $2,000, reaching a utility of about 25 (one quarter as valuable as the target) at $400,000. Utility then rises quickly, to 50 (half as valuable as the target) at $500,000 and 75 (three-quarters as valuable as the target) at a little less than $600,000. Utility then rises slowly to the target and beyond, stopping at 110 at $1,200,000. This curve indicates that utility might not rise above $1,200,000 (which might be the maximum allowed against that flag in the Red Team attack). Similar curves might be specified for the other Red Team flags. For example, Table 5-1 shows some hypothetical points that might be specified as points on the utility curves for the six Red Team flags. The table is read as follows. The table shows points on the utility functions for the six flags. The first two rows show points for the first flag, delay in replanning. The first row is the Red Team Work Factor (RTWF) in thousands of dollars to achieve the flag. The second row is the corresponding utility. These rows show a utility of 0 for $20,000, a utility of 12 for $200,000, a utility of 40 for $400,000, a utility of 80 for $600,000, a utility of 100 for $800,000, and a utility of 115 for $1,200,000. Utility for work factors between the values shown are determined by linear interpolation. The values shown are strictly hypothetical; an actual analysis would record the utility functions of the decision maker or evaluator.

Table 5-1. Points on the utility curves for Red Team flags (RTWF in thousands of dollars).

RTWF 1	20	200	400	600	800	1200			
U 1	0	12	40	80	100	115			
RTWF 2	20	200	300	450	500	600	700	800	1200
U 2	0	9	18	38	50	78	93	100	115
RTWF 3	40	400	800	1200					
U 3	0	60	100	125					
RTWF 4	40	240	400	480	800	1200	2000		
U 4	0	3	10	15	50	80	100		
RTWF 5	40	800							
U 5	0	100							
RTWF 6	20	520	800						
U 6	0	50	100						

In an actual analysis, these types of utility functions would be specified at all bottom-level attributes. This might result in some utility functions specified over metric values for some attributes, such as the flags, and some utility functions specified over qualitative descriptors of levels or degrees of attainment (e.g., high, medium, low) for other attributes.

5.2 Establish Weights for Red Team Attributes

The next step is to establish tradeoffs across attributes. In an additive MAU analysis, tradeoffs are expressed as weights. The weights are used to compare utilities across attributes and to combine the utilities on single attributes into utilities on groups of attributes. When assessing weights, one compares the importance of the swing on one attribute scale with the swing on another attribute scales. We will describe the commonly used reference comparison method for assessing weights in MAU analyses in this section.

In the Section 5.1, a utility function was developed for each Red Team attribute. This function assigned a utility to each possible level of each attribute. However, each utility function was defined independently of all others, so the resulting utilities are not directly comparable. Some attributes may be more important than others, and a measure of the priority, or relative importance, of each attribute is needed. In an additive MAU analysis, this is accomplished through a weighting system. As with the utility functions, weights reflect value judgments, and different decision makers (or different organizations) could have different weights. It is important to note that such “personal” selections may be based on requirements imposed by the customer or requirements of the application or the decision maker’s experience or the experience of the organization.

At each node in the hierarchy of attributes, the decision maker is first asked to rank order the importance of the swings in the attributes. One way to do this is to hypothesize a system that has the lowest level of all attributes and to specify the one attribute that should be increased to its goal level to provide the most improvement in the system. Consider the subattributes of Red Team Security. Assume a hypothetical UltraLog where it cost the Red Team $20,000 to achieve Flag 1, $20,000 to achieve Flag 2, $40,000 each to achieve Flags 3, 4, and 5, and $20,000 to achieve Flag 6. The decision maker in our example might judge that the greatest increase in security would be attained if Flag 4 (Security State) were increased to its goal level of $2,000,000. If this were done, the decision maker might then judge that the next-greatest improvement in security would be provided by increasing Flag 1 (Replanning) to its goal level of $800,000. The next-greatest increase in security might then be provided by increasing either Flag 2 (Data Availability), Flag 3 (Incorrect Data), or Flag 6 (Unauthorized Privileges). The least improvement might come from an increase in Flag 5 (Non-ALP Neighbor). These judgments imply that Flag 4 should have the greatest swing weight among the attributes of Red Team Security followed by Flag 1, Flag 2, 3, or 6, and finally Flag 5.

Next, a value of 100 is assigned to the most important swing, on Flag 4, and the decision maker is asked to specify how important the other swings are in determining security relative to the swing on Flag 4. Suppose in our example that the swing from $20,000 to $800,000 on Flag 1 were judged to increase security by half of what would be provided by an increase from $40,000 to $200,000 on Flag 4. Then a swing weight of 50 would be assigned to Flag 1. Suppose further that the increase from $20,000 to $800,000 to achieve Flag 2 were judged to increase security by one quarter of what the swing from $40,000 to $200,000 on Flag 4 would provide. Then Flag 2 would be assigned a swing weight of 25. Since the swings on Flags 2, 3, and 6 were judged to be equal, Flags 3 and 6 would also be assigned swing weights of 25. Finally, if the swing on Flag 5 were judged to contribute one tenth of the swing on Flag 4, then Flag 5 would be assigned a swing weight of 10. When these swing weights are normalized to add to 1.00, the results are weights of 0.21 for Flag 1, 0.11 for Flag 2, 0.11 for Flag 3, 0.43 for Flag 4, 0.04 for Flag 5, and 0.11 for Flag 6 (rounded to the nearest 0.01).

In a complete analysis, similar assessments would be made at each node in the attribute hierarchy. This allows evaluations to be rolled-up to each node including the top one. For our example, we will illustrate the method for evaluations of Red Team attributes only, so the weights given above are sufficient.

5.3 Red Team Evaluation of UltraLog Systems

The next step in our example is to specify or estimate the performance of the system design or implementation under evaluation. For purposes of the analysis, this means determining how each system would perform against each attribute. The basis for this determination could come from any of a number of sources. Assessments for some attributes might be made by examining the features of the design. Other attributes, such as cost, might be assessed using an estimating tool. Other attributes might be assessed using engineering judgment. Other assessments might be made using experiments or tests. Red team assessments fall into the last category, and this is the assessment that we will illustrate here.

Several aspects are important in the Red Team evaluations of UltraLog. One is the question of minimal required performance against any of the flags. This question is addressed by the setting of the individual utility curves for the attributes as described in Section 5.1. The “zero utility” level of each attribute specifies the minimum acceptable performance against that attribute, such that if any system fails to achieve that level on any flag, then that system is unacceptable. A second question is the performance goals against the flags. This question is also addressed by the utility functions. The goal level on each attribute is specified as the 100 utility level. A system that meets every goal receives a score of 100 on each attribute and a utility of 100 overall. A system could also receive an overall utility of 100 if it missed the goal on some attribute but exhibited superior performance on another attribute. The utility functions and weights should be established in such a way that systems that receive the same utility are equally attractive. A third question relates to the "baseline" evaluation. One baseline could be the minimally acceptable system that had a utility of zero. Another baseline could be an actual system. In this case, the baseline evaluation would be the Red Team evaluation of the specified baseline system. Another aspect of the evaluation is to show change over time. This is provided by completing Red Team evaluations of the different versions of UltraLog as it evolves over time. For example, UltraLog as it is developed in 2001 (UltraLog 01), 2002 (UltraLog 02), and 2003 (UltraLog 03). These evaluation can be compared, overall and on a attribute-by-attribute basis, to the minimally acceptable system, the baseline, and the goal. The following example illustrates this type of analysis for hypothetical Red Team evaluations of a baseline, UltraLog 01, UltraLog 02, and UltraLog 03.

Table 5-2 shows hypothetical evaluations of a baseline and three UltraLog systems, which might result from an analysis of the baseline and three instances of UltraLog. The results of the hypothetical Red Team attacks, in dollars of RTWF to achieve each flag, are shown for each system and for a minimally acceptable system and a goal system.

Table 5-2. RTWF (thousands of dollars) to achieve each flag for each system.

Attributes	Baseline	UltraLog 01	UltraLog 02	UltraLog 03	Minimal	Goal
Flag 1	20	400	600	950	20	800
Flag 2	20	300	800	1000	20	800
Flag 3	80	800	800	1100	40	800
Flag 4	20	400	400	1800	40	2000
Flag 5	240	240	600	600	40	800
Flag 6	70	520	800	800	20	800

In an additive MAU analysis, alternatives are evaluated by calculating a weighted multiattribute utility. The weights are the normalized swing weights that were assessed and calculated as explained in Sections 2.3 and 5.2. These weights are multiplied by the utilities of the alternatives on the attributes. These utilities are determined from the utility functions that were assessed as explained in Sections 2.2 and 5.1. In more complicated models, where the values of attributes interact (where how much you care about one attribute depends on the levels of other attributes), a model other than the additive model described here may be more appropriate. Keeney and Raiffa (1976) describe a number of such models (multiplicative, multilinear, and others) and discuss the conditions under which each should be used. This example assumes an additive model.

Table 5-3 shows the utilities for the alternative systems against the Red Team flags. These are calculated by applying the utility functions described in Table 5-1 to the performance, in RTWF, shown in Table 5-2. As described above, utility is taken as linear between the points given in Table 5-2. For example, consider the column for UltraLog 01. The $400,000 of RTWF that it took to achieve Flag 1 has a utility of 40, the $300,000 for Flag 2 has a utility of 18, the $800,000 for Flag 3 has a utility of 100, the $400,000 for Flag 4 has a utility of 10, all as entered on Table 5-1. The $240,000 for Flag 5 is between the two entries, a utility of 0 at $40,000 and a utility of 100 at $800,000. Thus, the utility for Flag 5 is:

(240 – 40)/(760) = 26. Notice that the utility for the baseline against Flag 4 is shown in parentheses. This is because the baseline is worse than the minimally acceptable system on this attribute.

Table 5-3. Utilities for each system.

	Weight	Baseline	UltraLog 01	UltraLog 02	UltraLog 03	Minimal	Goal
RED TEAM		**(2)**	**32**	**56**	**101**	**0**	**100**
Flag 1	0.21	0	40	80	106	0	100
Flag 2	0.11	0	18	100	106	0	100
Flag 3	0.11	7	100	100	119	0	100
Flag 4	0.43	(0)	10	10	95	0	100
Flag 5	0.04	26	26	74	74	0	100
Flag 6	0.11	5	50	100	100	0	100

The Red Team security evaluation is calculated as a weighted sum of the utilities for the flags, with the weights as explained in Section 5.2. For example, evaluation of UltraLog 01 is calculated as follows:

$$(0.21)(40) + (0.11)(18) + (0.11)(100) + (0.43)(10) +(0.04)(26) + (0.11)(50) = 32.$$

(All calculations are rounded for display to two decimal places for swing weights and to whole numbers for utilities.) The following paragraphs explain how these results can be used by system developers and evaluators.

What is the evaluation of the current UltraLog development? For purposes of this example, all evaluative discussions are limited to Red Team evaluations. This is an evaluation that has interest in its own right and could contribute to a larger evaluation that considers more attributes. The multiattribute utility Red Team evaluations are shown in the top row of Table 5-3. As a particular version of UltraLog is attacked by the Red Team, its performance can be evaluated as described above. The table shows, for example, that UltraLog 01 has an evaluation of 32. This evaluation is interpretable as a level of security against Red Team flags on a scale where 0 corresponds to a situation in which performance against each flag is at the minimum acceptable level, and 100 corresponds to a situation in which performance against each flag is at the goal level. The utility functions were constructed to be cardinal scales so the evaluation numbers carry more information than a simple ordering, and this information is useful for answering the next questions.

How does this compare with the requirements and goals? The goal represents a hypothetical system that contains the goal level of security on each flag. It has a utility of 100 for each flag, and its overall utility is 100. The security evaluations of the systems can be compared with the goal. Table 5-3 shows that UltraLog 01 meets the goal for Flag 3, but falls short for all of the other flags. Its overall evaluation of 32 indicates that it is 32% of the way from a minimally acceptable system to the goal.

How does this compare with the baseline? Table 5-3 shows that the baseline has an overall score of 2. However it also shows that the baseline performs below the minimal acceptable level on Flag 4. Thus, UltraLog 01 offers an improvement over the baseline by both meeting every minimal requirement and by improving to 32% of the goal.

How are the evaluations changing over time? UltraLog 01, UltraLog 02, and UltraLog 03 are subsequent versions of UltraLog. An examination of their evaluations shows how evaluations are changing over time. Figure 5-3 displays this information graphically and shows a steady progress toward the overall goal. The baseline does not meet every minimal requirement and its overall evaluation is also very close to minimally acceptable. UltraLog 01 improves to 32% of the goal, UltraLog 02 improves to 56% of the goal, and UltraLog 03 exceeds the goal by 1%.

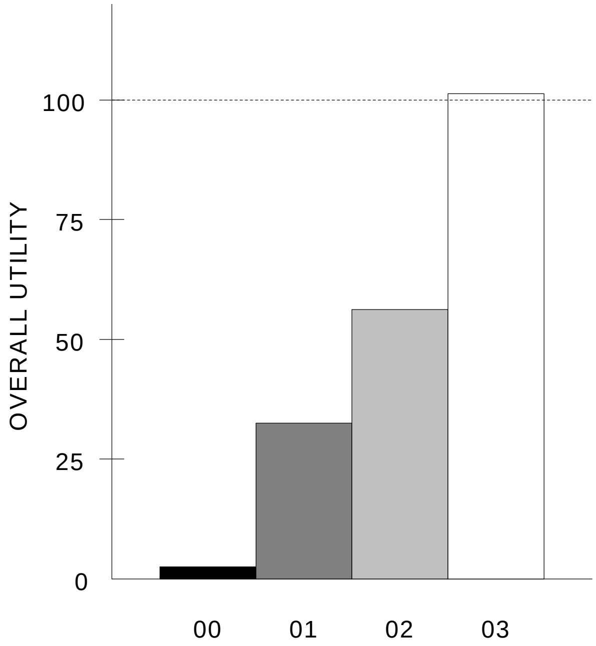


Figure 5-3. Overall Red Team evaluations.

What attributes need to be enhanced to improve the Red Team security evaluation? Any subattribute with a utility of less than the maximum could be improved to improve the Red Team security evaluation. Even UltraLog 03, which exceeds the goal on overall Red Team security, could be improved by improving in some areas, including in Flag 4 and Flag 5, where it falls short of the goal. The analysis shows that there is much room for improvement in UltraLog 01 and UltraLog 02. UltraLog 01 is high on Flag 3 but low on the other four flags. UltraLog 02 is high on Flags 2 and 3, moderately high on Flags 2 and 5, and low on Flag 4. Figure 5-4 shows how the utilities of the systems for the flags can be displayed graphically to show the room for improvement changes over time. This display also highlights that UltraLog 03 falls well short of the goal on Flag 5 even though it achieves an overall evaluation of over 100. UltraLog 03 exceeds the goal for Flags 1, 2, and 3, comes very close to the goal for Flag 4, and meets the goal for Flag 6. When considering the relative importance of achieving the goals, the net is that UltraLog 03 is better than the goal.

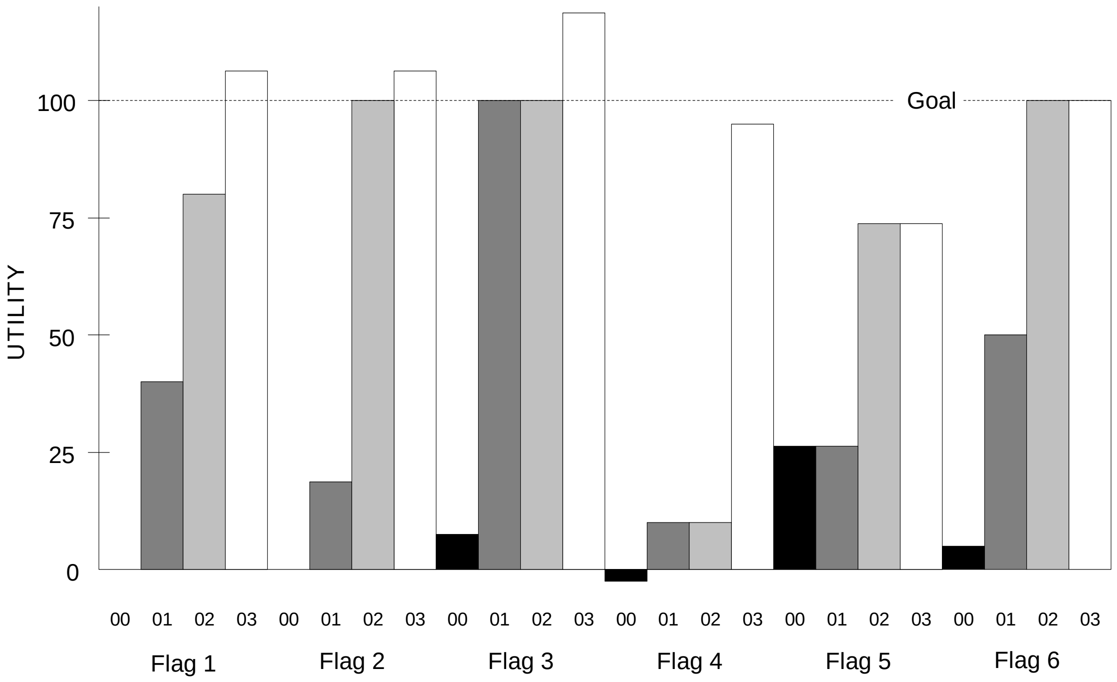


Figure 5-4. Evaluations against flags.

6. ELICITATION OF ULTRALOG UTILITY FUNCTIONS

After the approaches and illustrative materials described in Sections 4 and 5 were developed and pre-tested and revised, the required utility functions and weights were assessed in structured meetings with experts. These efforts sometimes resulted in it becoming apparent that changes were needed in the hierarchies of attributes themselves. Those changes were made during the course of the elicitations.

A near-final hierarchy and a set of utility curves and weights was determined during multiday group meetings. This Section summarizes work performed on the elicitation of utility curves and weights at the MAU Summit meeting at Schafer Corp on August 6-7, 2002, plus additional assessments made at the TIC on August 16.

In summary:

- Respondents
    - Logistics Experts (Retired military logistics officers)
    - Computer Security Experts
- Series of Meetings
    - Face-to-face facilitated meetings of 3-6 experts
    - Assessed utility functions and weights
    - Different groups assessed different inputs
- Variety of Elicitation Methods
    - “Balance Beam” for paired-comparisons
    - “Pricing Out”
    - Direct elicitation of “Swing Weights”

Participants. Participants at the MAU Summit included logistics experts and computer security experts. The following persons participated in the August 6-7 sessions:

- Marshall Brinn
- Jim Chinnis
- Mike Dyson
- Mark Greaves
- Rich Lazarus
- Leo Pigaty
- Tony Rozga
- Martin Solum
- Jake Ulvila
- Jim Workman

Preliminaries. It was decided to accept the argument that assessment of survivability requires a comparison of stressed performance with baseline performance. Essentially, this is due to the lack of a means to determine correctness of a logistics plan apart from using the baseline UL-generated plan as a standard. This argument has a substantial effect on the MOP definitions and scales.

Nomenclature. Parts of the MAU model that were assessed are described in the remainder of this Section. Note that two types of attribute identifiers are used. One is based upon the current set of MOEs and MOPs. These employ dashes between numbers, such as “MOP 1-1-1”. The other type of attribute identifier employs periods between numbers, such as “1.1.1.2 Supply Completeness”. Where appropriate, each assessed attribute is prefaced by its outline designation in the overall UltraLog Survivability MAU hierarchy as shown in Table 6-1. Note that the hierarchy in Table 6-1 reflects a considerable number of changes made both in the course of preliminary testing of the hierarchies developed and described in Sections 3, 4 and 5.

Table 6-1. Survivability MAU Hierarchy with Outline Designation Codes

1 Capability
1.1 Log Plan
1.1.1 Complete Plan Elements
1.1.1.1 Transport completeness
1.1.1.1.1 Transport completeness +7
1.1.1.1.2 Transport completeness +180
1.1.1.2 Supply completeness
1.1.2 Correct Plan Elements
1.1.2.1 Transport correctness
1.1.2.1.1 Transport correctness +7
1.1.2.1.2 Transport correctness +180
1.1.2.2 Supply correctness
1.1.3 Complete Info Display
1.1.3.1 Complete Trans Info Display
1.1.3.1.1 Transport info display completeness +7
1.1.3.3.2 Transport info display completeness +180
1.1.3.2 Supply info display completeness
1.1.4 Correct Info Display
1.1.4.1 Correct Transport info display
1.1.4.1.1 Transport info display correctness +7
1.1.4.1.2 Transport info display correctness +180
1.1.4.2 Supply info display correctness
1.2 Confidentiality & Accountability
1.2.1 Memory Confidentiality
1.2.1.1 % memory elements
1.2.1.2 Effort for memory
1.2.2 Disk Confidentiality
1.2.2.1 % Disk elements
1.2.2.2 Effort for disk
1.2.3 Transmission Confidentiality
1.2.3.1 % transmitted
1.2.3.2 Effort for transmitted
1.2.4 Accountability
1.2.4.1 % counter to policy
1.2.4.2 Effort to counter policy
1.2.4.3 % actions unrecorded
1.2.4.4 Effort to prevent recording
3 System Performance
3.1 Compute Log Plan
3.2 Replan
3.2.1 Replan time with Baseline <20sec
3.2.2 Replan time with Baseline >20sec
3.3 Assemble Data
3.2.1 Assemble data with Baseline <20sec
3.2.2 Assemble data with Baseline >20sec

Graphically, the upper-level hierarchy is shown in Figure 6-1:

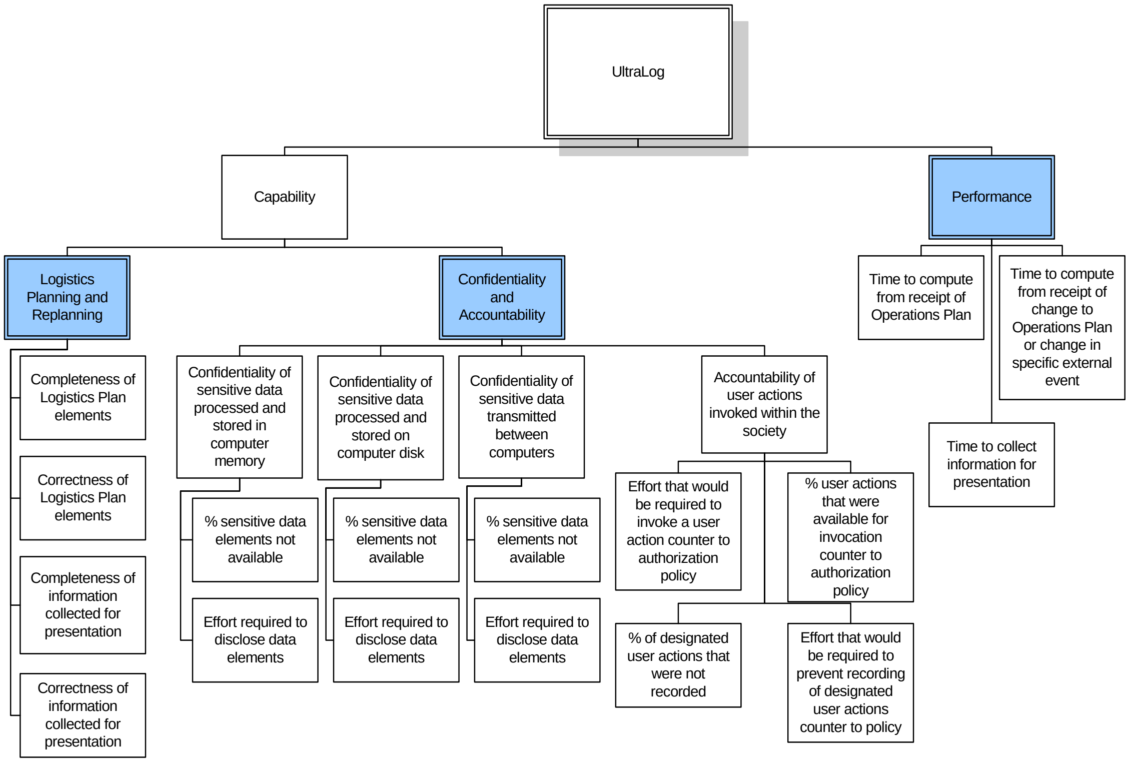
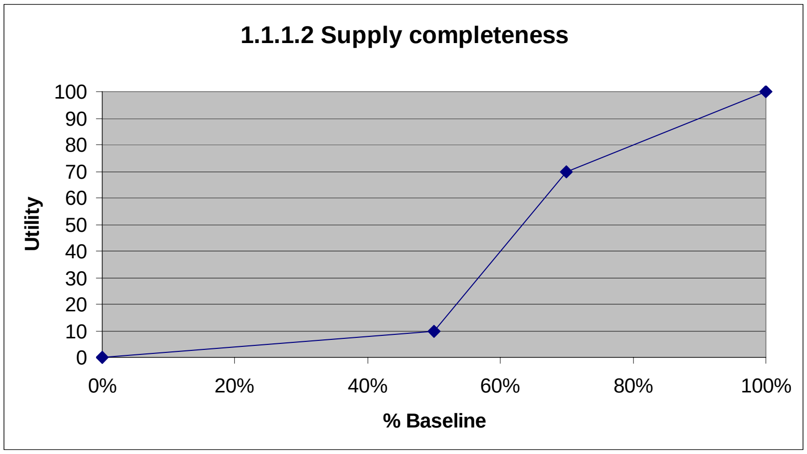

UltraLog
Capability
Performance
Logistics Planning and Replanning
Confidentiality and Accountability
Time to compute from receipt of Operations Plan
Time to compute from receipt of change to Operations Plan or change in specific external event
Time to collect information for presentation
Completeness of Logistics Plan elements
Correctness of Logistics Plan elements
Completeness of information collected for presentation
Correctness of information collected for presentation
Confidentiality of sensitive data processed and stored in computer memory
Confidentiality of sensitive data processed and stored on computer disk
Confidentiality of sensitive data transmitted between computers
Accountability of user actions invoked within the society
% sensitive data elements not available
Effort required to disclose data elements
% sensitive data elements not available
Effort required to disclose data elements
% sensitive data elements not available
Effort required to disclose data elements
Effort that would be required to invoke a user action counter to authorization policy
% user actions that were available for invocation counter to authorization policy
% of designated user actions that were not recorded
Effort that would be required to prevent recording of designated user actions counter to policy

Figure 6-1. Survivability MAU Hierarchy

6.1 Utility and Weight Elicitations from the MAU Summit

A rough description of the work accomplished during the principal elicitation sessions (August 6 and 7) is presented here, including approximate snippets of the very extensive and sometimes overlapping and simultaneous discussions that took place. Accuracy, either of the snippets of the discussion or the attributions given here, cannot be assumed.

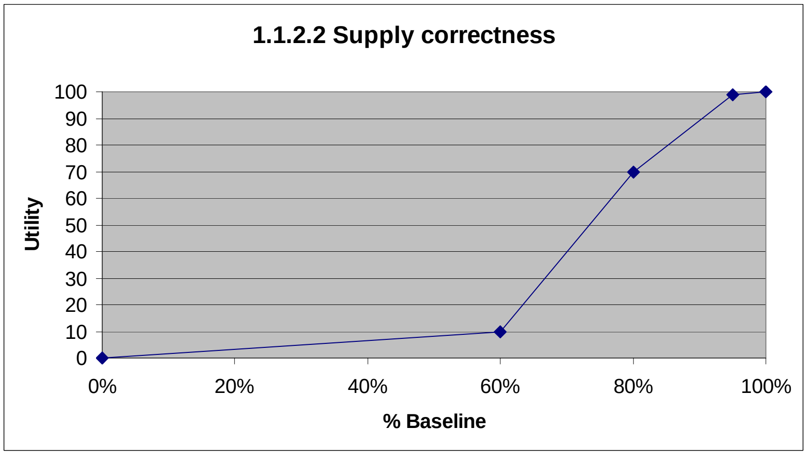
1.1.1.2. MOP 1-1-2 Supply Completeness



Assessed (x,y) points: (0,0), (50,10), (70,70), (100,100).

Rozga argued that 65-70% was the bottom, that “if a guy brought in a plan with 20% I’d throw him out.” Ulvila asked how valuable 50% completeness was—was it really as bad as nothing at all—and experts settled on a utility of 10.

1.1.2.2. MOP 1-2-2 Supply Correctness



Assessed (x,y) points: (0%,0), (60%,10), (80%,70), (95%,99), (100%,100).

Pigaty: “When you are 60 days before deployment, you want accurate data.”

Rozga: “TRANSCOM can only use the exact plan 48 hrs in advance. But if they know tonnage 15 days and out, they can get the planes in. Timing is very important here, and certain areas are more important. Same curve as for Completeness looks about right. Level 6 within 48 hours and Level 2 before.”

Pigaty: “In operations, there’s no distinction between complete and correct. I vote for same curve. But I think the drop is more at 80-85% correct gives 70 utility.”

Rozga: “If I really wanted to use UltraLog, I’d throw out the old systems, and [then the] correctness requirement becomes critical. We’ve built in all these layers, but a lot of them will go. If you’re talking about platforms, you really need 90%. I don’t see how you can get by with less than near perfect. It’s a JIT world.

“You aren’t changing the way the support system is structured. Unless you change those, you can’t replace parts of GCSS with UltraLog.”

Greaves: “Must be consistent with PAD requirements.”

Weight Elicitation: MOP 1-1-2 Supply Completeness vs MOP 1-2-2 Supply Correctness

Rozga: “I’d rather have completeness than correctness. If you’re cut off, would you rather get some stuff or nothing? Something even if incorrect. I will be making continual adjustments anyway. I will change my plan.”

To begin a search for a rate of exchange between Supply Completeness and Supply Correctness utilities, Ulvila posed the following choice represented on a balance beam:

Completeness:	**65**	70
Correctness:	**90**	70

^

Consensus was for the left option over the right (unanimous preference indicated by the boldface).

Workman: “There are two kinds of correctness: having confidence in the number means correctness is more important. [There are] not quite right answers versus injected nonsense.”

Brinn: “We really want the total number of correct answers you got. That’s all.”

Greaves: “We want to drive toward having a small, accurate plan.”

Pigaty: “The operator won’t know completeness unless a brigade is missing. If scattered, he won’t know.”

[?]: “The gaps really will be in blocks.”

Completeness:	**60**	90
Correctness:	**100**	60

^

Completeness:	**60**	90
Correctness:	**100**	70

^

Workman: “I want to know what I can trust and make up the rest.”

	Workman	Pigaty, Rozga
Completeness:	60	90
Correctness:	98	70

^

Workman: “I have 10 storerooms and have done nine inventories. With 70% correct, got to do it again.”

Completeness:	60	90	**<<<Indifference (Workman)**
Correctness:	95	85	
	^		

Completeness:	60	90	**<<<Indifference (Pigaty, Rozga)**
Correctness:	95	70	
	^		

Completeness:	50	90	**<<<Indifference (Dyson, Lazarus)**
Correctness:	95	70	
	^		

[Disagreement over whether we can tell what is missing.]

Comparison of implied swing weights (inversely proportional to utility swings on attributes with indifference comparison) led to

Implied swing weights:

	Workman	Pigaty/Rozga	Dyson/Lazarus
Completeness:	0.29	0.55	0.43
Correctness:	0.71	0.45	0.57

Seeking consensus, discussion resulted in shifts in weights to the following agreed set:

Completeness: 0.40

Correctness: 0.60

Transport attributes. The group considered what the differences are between Supply and Transport attributes.

Rozga: “Completeness is now more important than Correctness. And Transport is more important than Supply.”

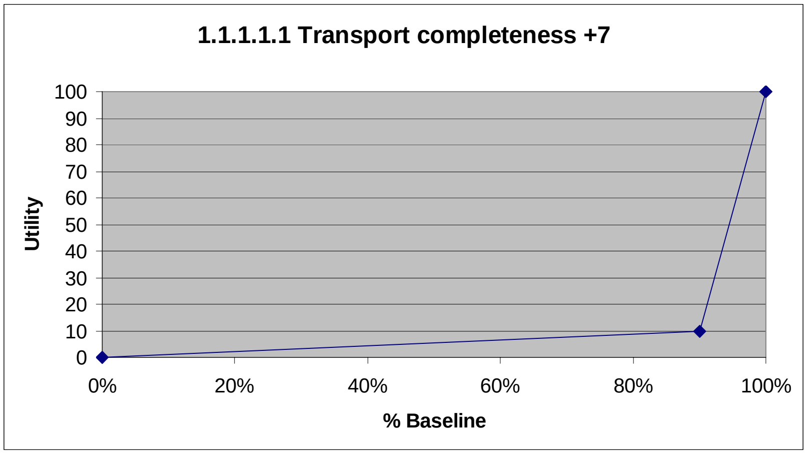
Ulvila asked if the Transport Completeness utility curve looked like the Supply Completeness curve. The group suggested that the curve was displaced to the right, remaining low utility longer.

Pigaty: “I can do stubby pencil calculations for Supply, but not for Transport. So, low completeness is more of a problem than low correctness. I can always roll more stuff on the next plane. I really need the planes.”

Lazarus: “If you get 70%, will you throw it out?”

After more discussion, Greaves decided to split the Transport attributes into two sub-attributes, one for <=7 days and one for 8-180 days.

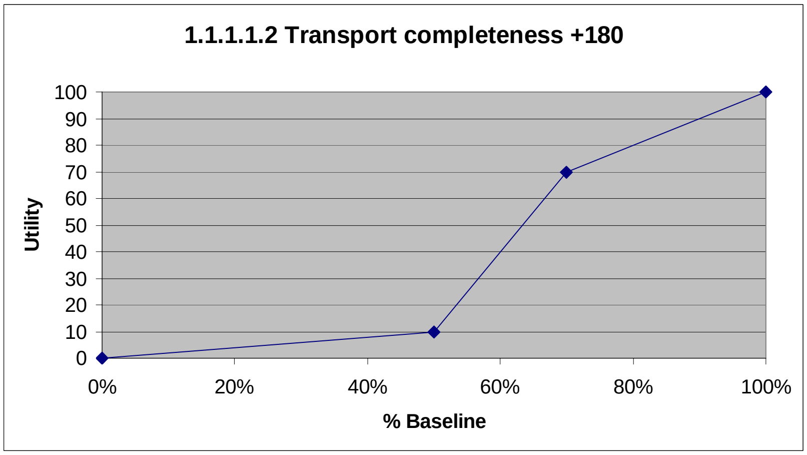
1.1.1.1.1 Transport Completeness D-+7



Assessed (x,y) points: (0%,0), (90%,10), (100%,100).

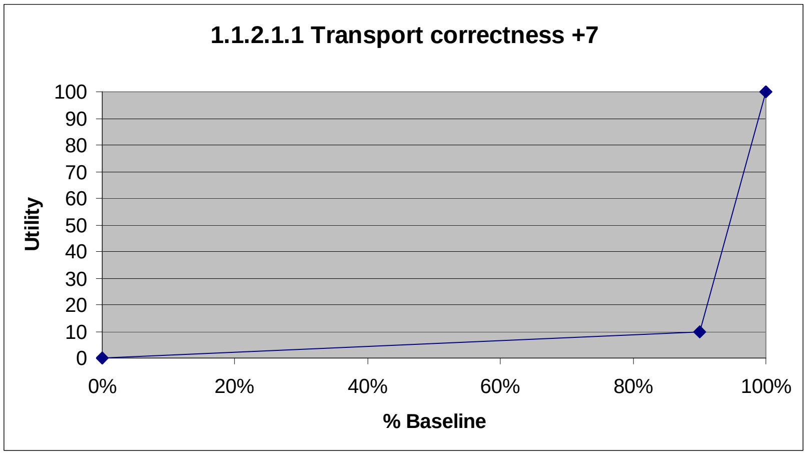
1.1.1.1.2 Transport Completeness +180 days

This was judged by the group to be the same as Supply Completeness:



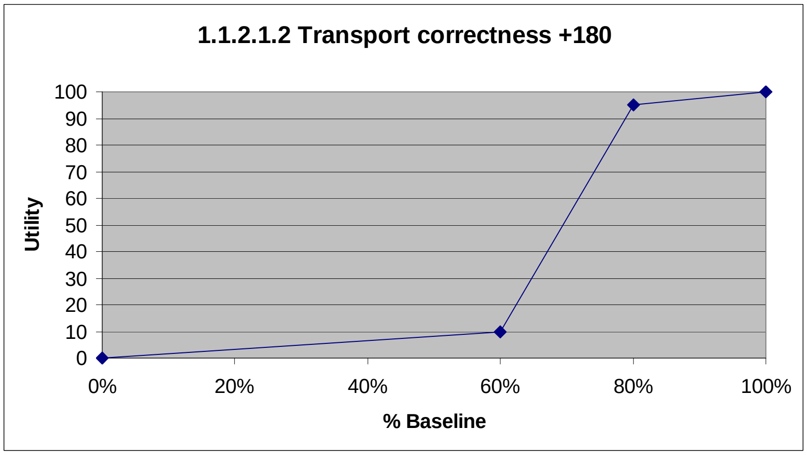
Assessed (x,y) points: (0%,0), (50%,10), (70%,70), (100%,100).

1.1.2.1.1 Transport Correctness +7 days



Assessed (x,y) points: (0%,0), (90%,10), (100%,100).

1.1.2.1.2 Transport Correctness +180 days



Assessed (x,y) points: (0%,0), (60%,10), (80%,95), (100%,100).

Dyson: “In out days, you’d want level 2 data.”

Weight Elicitation: MOP 1-1-1 Transport Completeness, Close-In vs. Far-Out

Close-in 90% to 100% swing in completeness set to 100. How does a swing on Far-out completeness from 50% to 100% compare in utility?

Rozga: “>50. The Far-out weight is more than half the close-in weight.”

+7:   90   95   <<<**Indifference (Rozga)**

+180:   100   80

^

+7:	90	95	<<<**Indifference (Pigaty)**
+180:	100	70	

^

Rozga: “3% means having no info about a tank battalion. 2% means having no info about a support battalion.”

After discussion, Rozga and Pigaty agree to:

+7:	95	93	<<<**Indifference (Pigaty & Rozga)**
+180:	50	70	

^

Implied swing weights:

Transport Completeness +7:	**0.85**
Transport Completeness +180:	**0.15**

Weight Elicitation: **MOP 1-2-1 Transport Correctness, Close-In vs. Far-Out**

Same as completeness.

Implied swing weights:

Transport Correctness +7:	**0.85**
Transport Correctness +180:	**0.15**

Weight Elicitation: **Close-in Transport, Completeness vs Correctness**

Workman and Pigaty have indifference points that imply equal swing weights. Rozga argues for closer to 60/40 favoring Completeness.

Agreed swing weights:

Transport Completeness +7:	**0.55**
Transport Correctness +7:	**0.45**

Weight Elicitation: **1.1.2.2 Supply Correctness vs 1.1.2.1.1 Transportation Correctness +7**

Transportation Correctness +7:	**99**	90	<<<**(Rozga)**
Supply Correctness:	**60**	80	

^

Transportation Correctness +7:	99	**90**	<<<**(Workman)**
Supply Correctness:	60	**80**	

^

Rozga: “In the first part of this period, supply is very important.”

Transportation Correctness +7: 99 95 <<<**Indifference (Workman)**

Supply Correctness: 60 70

^

Workman: “On the left, 99% of personnel will arrive as planned,but only 60% will have food, equipment.”

Rozga: “I still favor the left choice. Transportation trumps supply”

Transportation Correctness +7: 99 95 <<<**Indifference (Rozga)**

Supply Correctness: 60 80

^

Transportation Correctness +7: 60 90 <<<**Indifference (Brinn)**

Supply Correctness: 99 80

^

[?]:”There is some supply even if none arrives. People arrive with some supplies in the short term.”

Workman: “You lay out a plan. If you execute a 90% plan its pretty good—planes break, etc.”

Rozga: “You are talking about execution, which is different. For the first 60 days in Desert Storm, nothing got there that wasn’t approved by Schwarzkopf.”

Transportation Correctness +7: 90 **100** <<<**(Rozga)**

Supply Correctness: 100 **60**

^

Transportation Correctness +7: **90** 100 <<<**(Workman)**

Supply Correctness: **100** 60

^

Ulvila: “This shows an impasse. We may have to reconsider shapes of Transportation and Supply Correctness utility curves.”

Transportation Correctness +7: 95 100 <<<**Indifference (Workman)**

Supply Correctness: 100 73

^

Transportation Correctness +7: 95 **100** <<<**(Rozga)**

Supply Correctness: 100 **73**

^

Discussion led to agreement to compromise and then revisit if very low scores occur on Transportation Correctness.

Agreed swing weights:

Transportation Correctness +7:	**0.55**
Supply Correctness:	**0.45**

Brinn: “80% Transportation Correctness gives you nothing.”

Brinn: “I’d prefer a utility curve for [Transportation Correctness +7] that is 0 at 80% and 70 at 90%.”

Rozga: “I’d give 0 at 90%, and 99 at 95%.”

***Weight Elicitations:* Plan Elements (MOP 1-1 and 1-2) vs Display (MOP 1-3 and 1-4)**

Brinn: “It takes a lot of effort to improve the blackboard, but only a little to improve the display correctness (better protection against red team). What about stale data being displayed? If that’s a factor, we have a real tradeoff.”

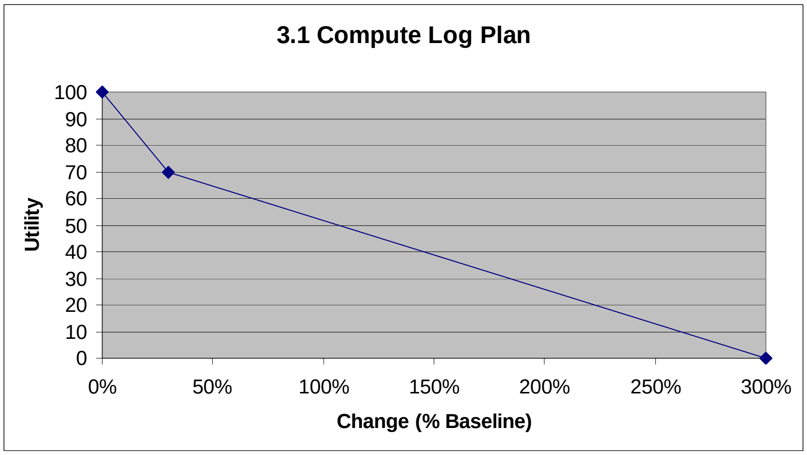
[?]: “One would expect the slower portrayal under stress. This is measured in a performance MOP.”

	Utility change	Weight (Greaves)
MOP 1-1 and 1-2 Plan Elements	70>>>90	**0.80**
MOP 1-3 and 1-4 Display	70>>>90	**0.20**

Solum & Brinn: “Data is not very volatile, thus low weight on display.”

(Agreed: utilities and weights between Supply and Transportation, near and far term, under MOPs 1-3 and 1-4 are the same as those corresponding items under MOPs 1-1 and 1-2.)

3.1 MOP 3-1-1 Time to Complete LogPlan



Assessed (x,y) points: (0%,100), (30%,70), (300%,0).

[?]: “This is like a big replan. When doing what-if analyses, timing is important.”

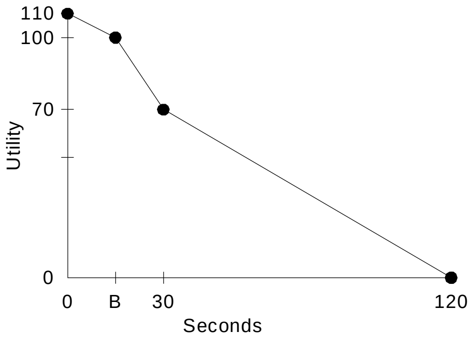
Rozga: “Plan in an hour is artificial.”

6.2 Additional Utility and Weight Elicitations

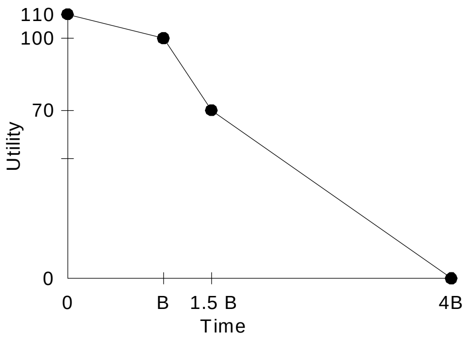
Further assessments were obtained on August 16 and are described in this section.

Utility Functions for Survivability MOPs 3-1-2 and 3-1-3

Utility functions for Survivability Measures of Performance 3-1-2 Time to Replan After a Specific External Event and 3-1-3 Time to Assemble and Portray Information After Query were agreed to on 16 August 2002 at the continuation of the Survivability MAU Summit. Both of these MOPs could have baseline response times ranging from fractions of a second to several (e.g., 30) minutes, depending on the type of actions requested of Ultra Log. These utility functions—unlike others—were assessed in terms of a combination of absolute stressed performance levels and as yet unknown level of baseline performance; thus they are plotted below in a somewhat different fashion than the other assessed utility curves:



If 0 < Baseline (B) < 20 seconds



If 20 seconds < Baseline (B)

For actions that take the baseline less than 20 seconds to perform, the utility function ranges over time up to two minutes. Utility is 100 if the stressed Ultra Log completes the action in the same time as the baseline. Utility drops to 70 if the response time is 30 seconds. The logistics experts stated that responses within 30 seconds were perfectly adequate. Utility drops to zero if response time is 120 seconds. At this level, system performance is substantially worse. Credit is given, up to a utility of 110, for response times less than the baseline.

For actions that take the baseline more than 20 seconds to perform, the group agreed to the bottom utility function, which is based on multiples of the baseline. Utility is 100 if the stressed Ultra Log completes the action in the same time as the baseline. Utility drops to 70 at 1.5 times the baseline and drops to zero at 4 times the baseline. Again, credit is given, up to a utility of 110, for response times less than the baseline.

These curves are added as attributes 3.2.1 and 3.2.2 under MOP 3-1-2 and attributes 3.3.1 and 3.3.2 under MOP 3-1-3. They would be combined using swing weights. The swing weights should represent the relative importance of the actions in each group. As a starting point, the group agreed to attach weights in proportion to the number of actions (queries or replans) that fall in each category (less than or greater than 20 seconds in the baseline). The determination of these numbers requires baseline runs of queries and replan requests.

It was also agreed that all or most of the same basket of queries used to assess MOPs 1-3 and 1-4 would be used to assess MOP 3-1-3. Jim Workman will define a set of Logistics Perturbations that will be used to assess MOP 3-1-2.

Notice that there is some possibility of a disconnect between the two curves. However, the 70 utility point is at 30 seconds on both curves if B=20 seconds. Furthermore, the 0 utility point is at 120 seconds on both curves if B=30 seconds. The disconnect problem arises for actions with baseline response times between 20 and 30 seconds. If there are many such actions, then we might want to revisit these curves and make slight adjustments.

Weight Elicitation: **Swing weights within MOPs 3-1-2 and 3-1-3**

Note that MOPs 3-1-2 and 3-1-3 are divided into actions taking the baseline less than or greater than 20 seconds. Weights between these divisions will be proportional to the number of actions in each class (<20 or >20).

Weight Elicitation: **Swing weights within MOE 3**

3-1-1 = 0.50

3-1-2 = 0.25

3-1-3 = 0.25

Consensus weight reflecting that retaining 60 minutes to develop a logistics plan is the central survivability claim. Rozga expressed the opinion that replanning is easy after a completed plan has been developed.

Weight Elicitation: MOE 1 vs. MOE 3

The combined weight for MOPs 1-1 and 1-2 = 40

The combined weight for MOPs 3-1-1 and 3-1-2 = 60

Time is usually emphasized over quality once minimum thresholds of acceptability have been met. No one expects a perfect plan, but one might be asked for, “An 80% plan by 2:00.”

Partial credit for Level 2 compared with Level 6 detail

We will go with “Level 2 is 80% of Baseline Level 6” for completeness and correctness. It is implemented by first determining the completeness or correctness of a plan element compared with baseline, then multiplying by 80% if only Level 2 detail is provided, and then entering the result as the abscissa (x-coordinate) and reading the utility.

Summary of Implied Weights

A chart displaying the normalized and rounded weights implied by the assessments is shown on the following page.

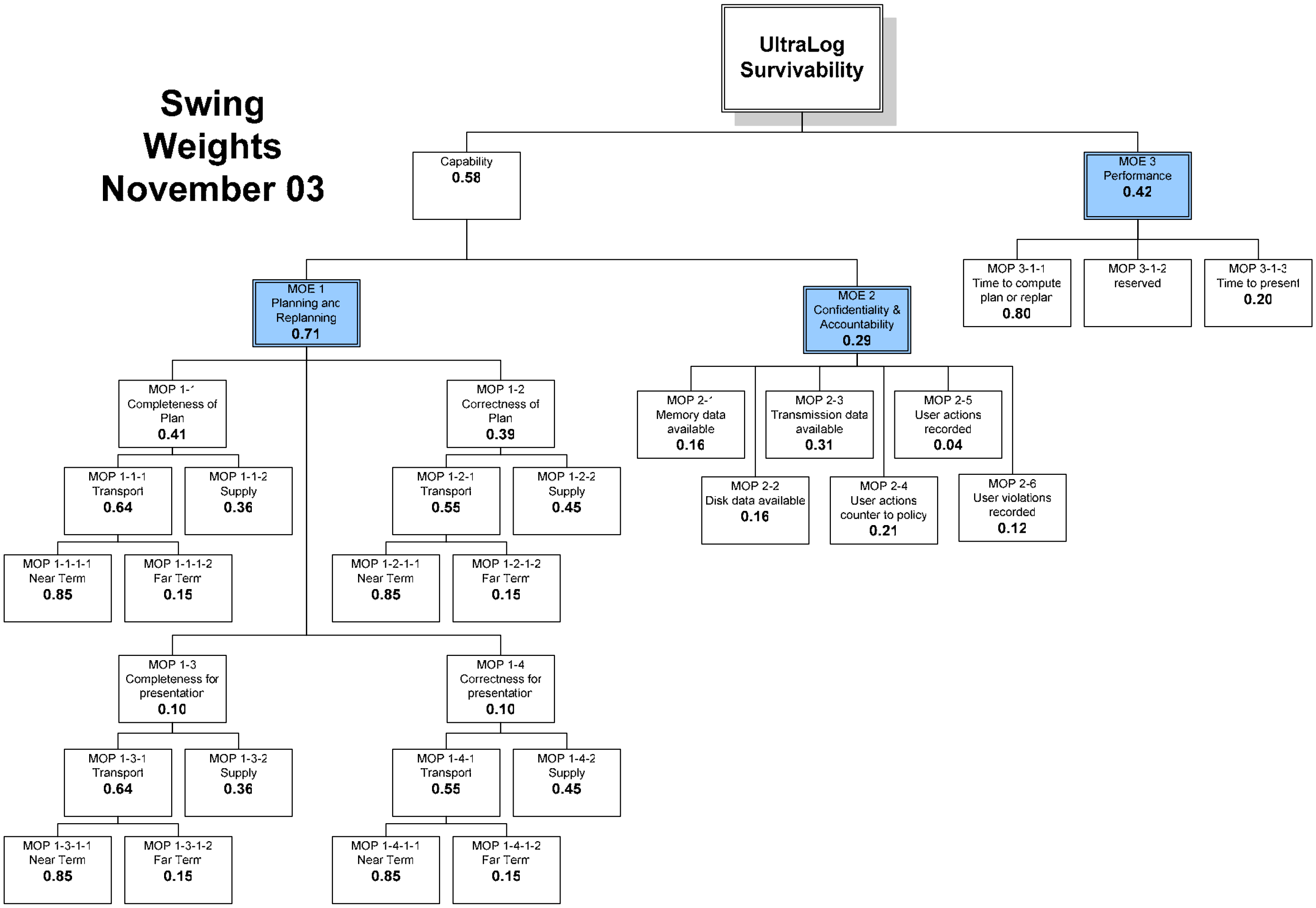

Swing Weights November 03
UltraLog Survivability
Capability 0.58
MOE 3 Performance 0.42
MOP 3-1-1 Time to compute plan or replar 0.80
MOP 3-1-2 reserved
MOP 3-1-3 Time to presenl 0.20
MOE 1 Planning anc Replanning 0.71
MOE 2 Confidentiality & Accountability 0.29
MOP 1-1 Completeness of Plan 0.41
MOP 1-2 Correctness of Plan 0.39
MOP 2-1 Memory data available 0.16
MOP 2-3 Transmission data available 0.31
MOP 2-5 User actions recorded 0.04
MOP 2-2 Disk data available 0.16
MOP 2-4 User actions counter to policy 0.21
MOP 2-6 User violations recorded 0.12
MOP 1-1-1 Transporl 0.64
MOP 1-1-2 Supply 0.36
MOP 1-2-1 Transporl 0.55
MOP 1-2-2 Supply 0.45
MOP 1-1-1-1 Near Term 0.85
MOP 1-1-1-2 Far Terrr 0.15
MOP 1-2-1-1 Near Term 0.85
MOP 1-2-1-2 Far Terrr 0.15
MOP 1-3 Completeness for presentatior 0.10
MOP 1-4 Correctness for presentatior 0.10
MOP 1-3-1 Transporl 0.64
MOP 1-3-2 Supply 0.36
MOP 1-4-1 Transporl 0.55
MOP 1-4-2 Supply 0.45
MOP 1-3-1-1 Near Term 0.85
MOP 1-3-1-2 Far Terrr 0.15
MOP 1-4-1-1 Near Term 0.85
MOP 1-4-1-2 Far Terrr 0.15

Figure 6-1. Hierarchy and Normalized Swing Weights for Survivability MAU Structure

6.3 Weights in the Absence of Near Term Baseline Transportation

In late 2003, testing revealed that baseline runs did not exhibit near term transportation tasks. After exploring the implications, DSA recommended a revision conditioned on the number of near term tasks remaining very low. Swing weights in the MAU model for this case were computed by zeroing out the weights for all Near Term Transport nodes and making corresponding reductions in the weights on their parent nodes (Transport). Based on the earlier elicitation procedure followed, weights at higher levels should remain the same. For example, the table below shows the entered weights, their calculated local weights, the new entered weights (if there are no near-term transport tasks), and the new local weights that change.

The new weights are only recommended if there are no near-term transport tasks or if the very small number of near-term transport tasks (e.g., on the order of 1% of the number in the initial base run) would indicate that the number should be rounded down to zero.

These are consistent with the way that tradeoffs were elicited. At the lowest levels, tradeoffs were elicited for completeness and correctness of supply and near-term and far-term transport. At other levels, tradeoffs were elicited for time to compute compared with completeness and correctness (combined) and for confidentiality and accountability compared with completeness and correctness (combined).

Table 6-2: Weights for the no near term transportation case

MOP	Title	Entered Wts	Local Wts.	New Entered Wts	New Local Wts
	Capability	43.3	.58		
	MOE 1: Planning & Replanning	30.9	.71		
1-1	Completeness of Plan	12.6	.41		
1-1-1	Transport completeness	8.1	.64	1.2	.21
1-1-1-1	Transport completeness Near Term	.85	.85	0	0
1-1-1-2	Transport completeness Far Term	.15	.15	.15	1.00
1-1-2	Supply completeness	4.5	.36		.79
1-2	Correctness of Plan	12.1	.39		
1-2-1	Transport correctness	6.7	.55	1.0	.15
1-2-1-1	Transport correctness Near Term	.85	.85	0	0
1-2-1-2	Transport correctness Far Term	.15	.15	.15	1.00
1-2-2	Supply correctness	5.5	.45		.85
1-3	Completeness for Presentation	3.1	.10		
1-3-1	Complete Presentation of Transport	.791	.72	.119	.28
1-3-1-1	Complete Transport Presentation Near	.85	.85	0	0
1-3-1-2	Complete Transport Presentation Far	.15	.15	.15	1.00

1-3-2	Complete Supply Presentation	.30	.28		.72
1-4	Correctness for Presentation	3.1	.10		
1-4-1	Correct Transport Presentation	.647	.59	.097	.18
1-4-1-1	Correct Transport Presentation Near	.85	.85	0	0
1-4-1-2	Correct Transport Presentation Far	.15	.15	.15	1.00
1-4-2	Correct Supply Presentation	.45	.41		.82
	MOE 2: Confidentiality & Accountability	12.4	.29		
2-1	Memory Data Available		.16		
2-2	Disk Data Available		.16		
2-3	Transmission Data Available		.31		
2-4	User actions counter to policy		.21		
2-5	User actions recorded		.04		
2-6	User violations recorded		.12		
	MOE 3: System Performance	30.9	.42		
3-1-1	Time to compute plan*		.40		
3-1-2	Time to replan*		.40		
3-1-3	Time to present		.20		

*Note: MOPs 3-1-1 and 3-1-2 have been combined, with Local Wt = 0.80

7. INFORMATION INFRASTRUCTURE

The objective of UltraLog is to demonstrate that large-scale, distributed, agent-based logistics C2 systems can survive under cyber and kinetic attacks. UltraLog's goal is to: "operate with up to 45% information infrastructure loss in a very chaotic environment with not more than 20% capabilities degradation and not more than 30% performance degradation for a period representing 180 days of sustained military operations in a major regional contingency." This and the following sections addresses what is meant by the phrase, "45% information infrastructure loss." This section describes the work performed prior to the full specification of the UltraLog system and how it could be degraded. The following Section (Section 8) describes a revised approach that was developed after the UltraLog experimental situation was better defined. The present Section is based on another DSA report submitted under the same contract: (Ulvila et al., 2001b).

The UltraLog infrastructure consists of four types of assets: processing and storage, local-area communications, wide-area communications, and external resources. We propose a method that starts by drawing a distinction between the loss of assets and the loss of information infrastructure, which is based on both the asset and the configuration of assets. We illustrate how information infrastructure loss in a configuration depends not only on the number and types of assets lost, but also on their relationships to other assets. Making a few simplifying assumptions, we describe a method for determining the information infrastructure loss that would result from the loss of each asset. The information infrastructure loss is shown to depend on the configuration that would remain if the asset were lost. The asset's information infrastructure loss value is first calculated predicated on its being the first asset lost. After one asset is lost, the information infrastructure loss value of each remaining asset is calculated assuming that each would be the next asset lost. The method is repeated for any number of assets lost.

We show that the determination of "45% information infrastructure loss" does not specify uniquely which assets are lost. Rather, there are many ways that asset losses can total 45% information infrastructure loss, even if each asset is worth the same individually. In two simple examples, we show how the loss of as few as two or as many as eleven of the eighteen assets in an illustrative configuration could result in approximately 45% information infrastructure loss.

The method that we present has some simplifications, but it illustrates the method. The loss calculation method does not have to be perfect to be useful, and the determination of what constitutes a 45% loss in information infrastructure does not have to be exact to be useful in simulated (Red Team) attacks or for determining the robustness of UltraLog to information infrastructure losses due to kinetic or information warfare attacks. Nevertheless, we also describe several ways that the method could be extended to provide more precise statements about information infrastructure loss. These extensions, however, come at the price of a more complex procedure. The method described here could be developed into a software tool that would be useful to the planners of experiments and assessments.

7.1 UltraLog Assets and Information Infrastructure

Four major types of assets combine to make up the UltraLog system: processing and storage, local-area communications, wide-area communications, and external resources. This section illustrates how assets combine into a configuration and how the information infrastructure loss associated with the loss of assets depends on both the assets lost and their relationships to the remaining assets.

7.1.1 UltraLog assets

For our purposes, we are considering Information Infrastructure to be those information-processing resources that UltraLog requires in order to run, but which are not UltraLog elements themselves (e.g., agents or PlugIns). We believe this categorization clearly distinguishes a focus on infrastructure assets.

The four asset types we have chosen to describe Information Infrastructure are a compromise between expressiveness and complexity. These assets can be either supplemented or subdivided if it proves that the asset types are inadequate. Additional asset types would not undermine the method, but could make the method harder to use.

There is a trade-off between expressiveness and complexity that we have tried to balance. We believe that our proposed asset types are expressive enough to be immediately useful and yet restrained enough to reduce complexity. The asset types that we propose are the following:

- Processing and storage;
- Local-area communications;
- Wide-area communications; and
- External resources.

A central aspect in how we categorize assets is the nature in which losses are likely to occur. It is usually the case that any kinetic or information warfare attack on UltraLog's information infrastructure is likely to result in the loss, in discrete units, of one or more assets of the types described above. We do not believe that specific attacks are likely to reduce capabilities in less than discrete units. Even denial-of-service attacks, for instance, are more likely to completely eliminate, rather than reduce, a capability (e.g., a communication link's bandwidth). Extensions of the methodology to handle reduced-capability cases without introducing unnecessary complexity are addressed in Section 7.2.4.

Processing and storage. From a computational viewpoint, it makes sense to treat the processing and storage elements of the infrastructure as a single asset type. Most "processing elements" within UltraLog are simply computers, which are a combination of processors, memory, and storage capabilities (ignoring communications aspects for the moment). Processing "power" can be thought of as a combination of processor capability and available memory. Similarly, available storage is usually a prerequisite for the computer to do work. If a failure or compromise in any of these resources occurs, the usual result is a comprehensive loss of the computer's capabilities. Even in cases where a "discretionary" loss occurs, such as a node's becoming untrustworthy, the loss will usually be "all-or-nothing." (Note that untrustworthy agents are not an infrastructure issue, because agents are part of UltraLog.) Therefore, although we acknowledge that the combination of processing and storage resources into a single asset is a simplification, we argue that it is also correct semantically.

Local-area communications. Local-area communications are important for UltraLog elements that must cooperate with other elements in the same geographical location. This definition intentionally excludes those that operate on the same node as well as those that must cooperate with remote elements. The processor asset addresses the former case, and the wide-area communications asset, discussed next, addresses the latter. Typically, local-area communications are located within a facility that is usually protected. This correlates well with particular types of attacks and types of countermeasures, facilitating common-mode analyses for tradeoffs within this asset type. For example, if local-area communications were discovered to be a weak link in terms of the effect of infrastructure loss, similar protection improvements could be applied across all geographical locations simultaneously.

Wide-area communications. Wide-area communications are important for UltraLog elements that must cooperate in the processing of the overall logistics plan over widespread geographical locations. This is an important asset area because sub-areas of UltraLog could become separated, isolating large groups of UltraLog elements from others. This is a distinct infrastructure area because similar attacks and methods could be applied specifically in this technology area.

External resources. It is important for UltraLog to capture some of its dependencies on the databases and other resources that it uses to develop a logistics plan, but which are not actually UltraLog components themselves. This is a distinct asset in that the external resources usually, or often, do not reside on the same node as UltraLog elements. These assets also have separate attack factors (e.g., non-UltraLog users and maintainers) and protection factors (e.g., separate facilities). These assets could become isolated from UltraLog via communications failures (local- or wide-area) that are independent from those that affect UltraLog elements directly.

7.1.2 An example UltraLog system

Any UltraLog system is configured from a group of four types of assets: processing and storage, local-area communications, wide-area communications, and external resources. Figure 7-1 shows an example of such a configuration of assets: 8 processing and storage assets (labeled P1 through P8), 6 local-area communications assets (labeled L1 through L6), 2 wide-area communications assets (labeled W1 and W2), and 2 external resources (labeled E1 and E2). The particular configuration shown is small enough to provide the basis for a completely contained illustration of the proposed method yet large enough to highlight nontrivial aspects of the interrelationship of assets and the configuration that determines information infrastructure loss.

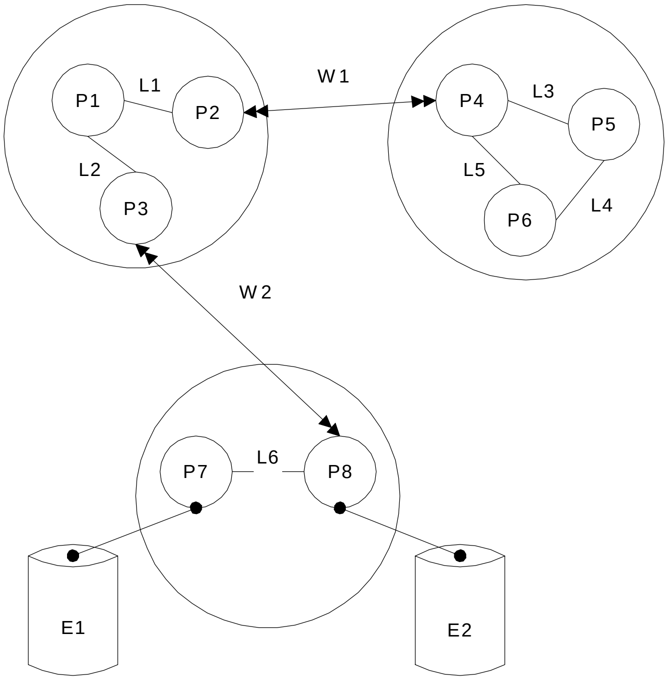


Figure 7-1. Configuration of assets.

If some assets were lost, is it possible to quantify easily that loss? Suppose that one were asked to specify the information infrastructure loss that would result from a loss of: 2 processing and storage assets, 3 local-area communications assets, and 1 external resource. One way to achieve this loss of assets is to eliminate or completely destroy P6, P7, L4, L5, L6, and E1. The resulting system configuration is shown in Figure 7-2. As shown, the system loses the two processing and storage assets, and it loses the local-area communication assets to which those processors (when we use the term, “processor,” we are referring to a processing and storage asset) were connected and the external resource that was used by one of the lost processors. This is a significant loss in the information infrastructure, but the remaining infrastructure retains both the local-area and wide-area communications with the remaining assets that were provided in the original configuration.

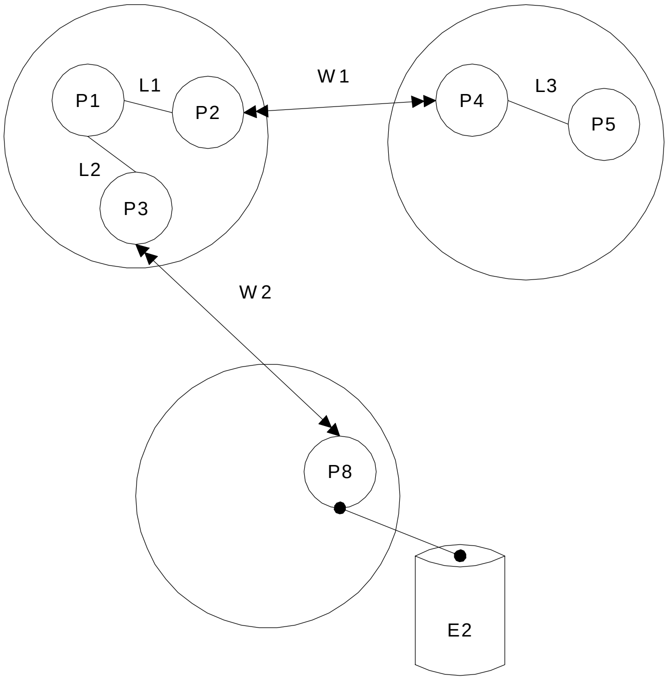


Figure 7-2. Loss of assets (first example).

Contrast this with the configuration shown in Figure 7-3. Figure 7-3 shows the configuration that results from a loss of P4, P8, L1, L2, L4, and E1. Figures 7-2 and 7-3 both show losses of 2 processing and storage assets, 3 local-area communications assets, and 1 external resource. So, in this respect, they both show the same loss in assets. However, in Figure 7-3, all of the remaining processors are isolated. Although each local- and wide-area communication link remains connected to a processor, there is nothing on the other end of the connection. The wire is still there, but only one side is connected. It is a “dangling wire.” The same is true of the remaining external resource. The asset is still there, but it is not connected to the system.

One could argue that more of the information infrastructure is lost in Figure 7-3 than in Figure 7-2. Even though the same number and types of assets remain, the remaining assets are connected in Figure 7-2 but not in Figure 7-3. As far as the information infrastructure is concerned, the remaining L3, L5, L6, W1, W2, and E2 are have lost all of their function, so they might as well have been destroyed. If this is the case, then we will say that going from the configuration in Figure 7-1 to that in Figure 7-2 involves the same loss in *assets* as going from Figure 7-1 to Figure 7-3. However, the *information infrastructure loss* is greater going to Figure 7-3 than going to Figure 7-2. An asset, even though remaining, is lost to the information infrastructure if it ceases to provide its function.

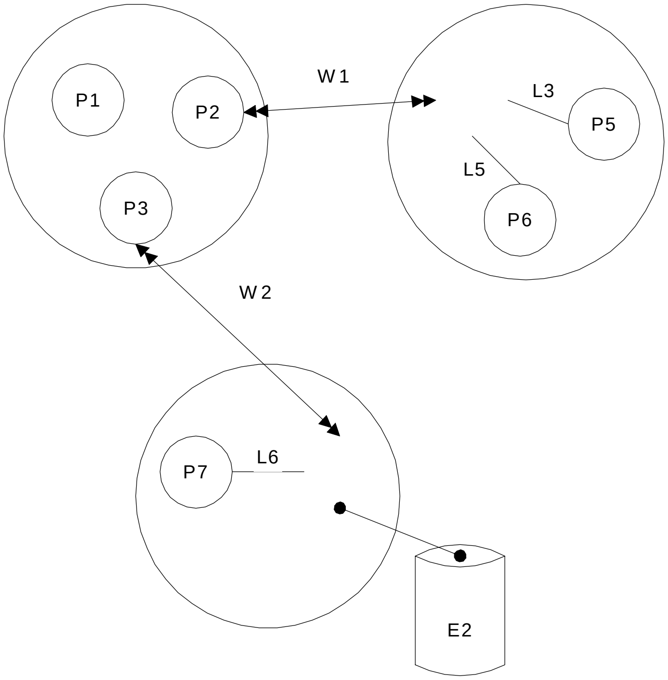


Figure 7-3. Loss of assets (second example).

These examples illustrate some of the difficulties in specifying the information infrastructure loss associated with the loss of assets and why one cannot simply say that a 45% information infrastructure loss corresponds to a loss of 45% of the assets in a system. Information infrastructure loss depends critically on which assets are lost and on the relationships of those assets to the assets that remain. One could not simply tell a Red Team that they are free to destroy (or remove) 45% of the assets or of each asset type. As the examples in this section show, one could do more damage by selectively removing the assets so as to incapacitate as much of the system as possible. In the example, one would first remove the processors that are connected to the most communication and external resources. Then one would remove assets that were not affected by the first removal. One could constrain the process by forcing some randomness in the removal of assets or to impose other rules that restrict the dispersal of the assets removed. However, one is still left with having to determine the extent of information infrastructure loss. One could define the amount in a way that is confounded with the selection process (e.g., by defining a 45% information infrastructure loss as the loss of a random 45% of each asset class). This has the additional problem of having to determine what is meant by “randomness” as well as “loss.” It also does not address the problem illustrated by Figures 7-2 and 7-3.

7.2 A Method for Determining Information Infrastructure Loss

This section presents our straw man method for determining information infrastructure loss. The method begins by defining the total value of information infrastructure as the sum of the values of all of the assets. Next, the information infrastructure loss associated with the loss of each asset is determined. This is the sum of the values of: the asset itself and the remaining assets that would be rendered useless by its loss. If an asset is lost, then the values of the remaining assets are determined in a similar manner after adjusting the configuration to account for the initial loss. The process is repeated to account for the loss of additional assets. The information infrastructure loss at any stage is calculated as the sum of the values of the assets lost, as calculated at the time that they were lost. This can be stated in percentage terms by dividing the loss by the total value of the information infrastructure.

This manner of computing different values as assets are lost is similar to the method used by Brown and Ulvila (1983) to determine the value of nuclear inspection activities. As activities were performed, the values of the remaining activities were adjusted to take account of the overlap between the activities performed and those remaining.

Section 7.2.1 presents the method and illustrates it for the example configuration in Figure 7-1 if each asset is considered to be of equal value. Section 7.2.2 repeats the analysis and illustration if different types of assets have different values. Section 7.2.3 repeats the analysis with a different rule for when an asset is considered lost. Section 7.2.4 presents some extensions to and uses of the method.

7.2.1 Illustration with equal value of assets

This section contains a complete description of an analysis of the information infrastructure loss associated with the loss of assets from the configuration shown in Figure 7-1 when each asset is of equal value. Figure 7-1 contains a configuration with 18 assets: 8 processing and storage assets (labeled P1 through P8), 6 local-area communications assets (labeled L1 through L6), 2 wide-area communications assets (labeled W1 and W2), and 2 external resources (labeled E1 and E2). Without loss of generality, we can assign a value of 1.0 to each asset, which gives a total value of 18.0 to the entire information infrastructure in the configuration.

Next, we will make the following assumptions with regard to the assets in the system. First that a processor retains its value even if the local- or wide-area communication asset to which it is connected is lost. Second, that a local- or wide-area communication asset has no value if it is connected to only a single processor (i.e., if it is a “dangling wire”). Third, that an external resource has no value unless it is connected to a processor.

Step 1: Calculate the information infrastructure loss associated with the loss of each asset. The first step is to calculate the information infrastructure loss that would occur if each asset were the only one lost. This is the sum of the value changes that would occur if the asset were lost. For example, if P1 were the only asset lost, one processor (P1) would be lost and two local-area communication assets (L1 and L2) would be rendered useless (because they would then be connected at only one end). This results in an information infrastructure loss of 3.0. Table 7-1 shows the initial information infrastructure loss for each asset.

Step 2: Adjust the calculation to account for the loss of an asset. If an asset were lost, then the value of the remaining assets might change. If P1 were lost, then the information infrastructure loss values of L1 and L2 would be reduced to 0.0. That is, the loss of these two “dangling wires” would not result in any additional information infrastructure loss. The information infrastructure losses associated with these two local-area communication lines were counted when they were rendered useless by the loss of P1, so no additional information infrastructure loss would be associated with the loss of these assets. The information infrastructure loss values of L1 and L2 are the only ones affected by the loss of P1.

Similarly, if P2 were the only asset lost, then its loss would reduce the information infrastructure loss of L1 and W1 to zero. The loss of P3 would reduce the values of L2 and W2 to zero. The loss of P4 would reduce the values of L3, L5, and W1 to zero. The loss of P5 would reduce the values of L3 and L4 to zero. The loss of P6 would reduce the values of L4 and L5 to zero. The loss of P7 would reduce the values of L6 and E1 to zero. The loss of P8 would reduce the values of L6, W2, and E2 to zero.

The loss of any single local-area communication asset (L1–L6), wide-area communication asset (W1–W2), or external resource (E1–E2) would reduce the values of all processors to which it is connected by 1.0.

Table 7-1. Information infrastructure loss for each asset (if the only one lost).

	Information Infrastructure Loss				
Asset	Processors	Local-area communications	Wide-area communications	External resources	Loss Value
P1	1	2	0	0	3.0
P2	1	1	1	0	3.0
P3	1	1	1	0	3.0
P4	1	2	1	0	4.0
P5	1	2	0	0	3.0
P6	1	2	0	0	3.0
P7	1	1	0	1	3.0
P8	1	1	1	1	4.0
L1	1	0	0	0	1.0
L2	1	0	0	0	1.0
L3	1	0	0	0	1.0
L4	1	0	0	0	1.0
L5	1	0	0	0	1.0
L6	1	0	0	0	1.0
W1	0	0	1	0	1.0
W2	0	0	1	0	1.0
E1	0	0	0	1	1.0
E2	0	0	0	1	1.0

Step 3: Calculate the information infrastructure loss for a group of assets. The procedure can be repeated to calculate the information infrastructure loss for combinations of assets. For each set of lost assets, one first determines which assets are either lost or rendered useless, and then the extent of the loss is calculated. Table 7-2 shows the results for a variety of asset losses, many of which result in approximately a 45% information infrastructure loss. For example, consider the second row and assume that the assets were lost in the order shown, P2, P4, and P5. The loss of P2 is an information infrastructure loss of 3.0 (from Table 7-1). Next, the loss of P4 is an information infrastructure loss of 3.0 (adjusted down from 4.0 in Table 7-1 because W1's information infrastructure loss was associated with the prior loss of P2). Finally, the loss of P5 is an information infrastructure loss of 2.0 (adjusted down from 3.0 in Table 7-1 because L3's information infrastructure loss was associated with the prior loss of P4). The total information infrastructure loss is: 3.0 + 3.0 + 2.0 = 8.0, which accounts for the cumulative information infrastructure loss of 3 processors (P1, P2, and P3), 4 local communications (L1, L3, L4, and L5), and 1 wide-area communication link (W1).

Table 7-2 shows eleven different combinations of asset losses that result in an information infrastructure loss of 8 of the 18 components (44.4%). Also shown are the information infrastructure losses for the two examples in Section 2. As expected, the assets lost in Figure 7-2 have an information infrastructure loss of only 33% compared with an information infrastructure loss of 67% for the assets lost in Figure 7-3. (Notice that the information infrastructure loss associated with the loss of a group of assets does not depend on the order in which the assets were lost. The value of each asset at the time that it is lost, however, depends on which assets were remaining at that time.)

As one can see, there are many ways to achieve a 44% information infrastructure loss. It can involve the loss of as few as two assets (P4 and P8) or as many as eight assets (L1-6 and E1-2). Table 7-2 shows that four different combinations of three assets and three combinations of five assets also produce 44% information infrastructure losses. The relationship between the number of assets lost and information infrastructure loss is far from unique.

Table 7-2. Information infrastructure loss for asset loss combinations (equal-valued assets).

	Information Infrastructure Loss					
Assets Lost	Processors	Local-area	Wide-area	External	Loss	% loss
P4, P8	2	3	2	1	8.0	44%
P2, P4, P5	3	4	1	0	8.0	44%
P1, P5, P6	3	5	0	0	8.0	44%
P1, P3, P8	3	3	1	1	8.0	44%
P5, P6, P7	3	4	0	1	8.0	44%
P1, P6, W2, E1	2	4	1	1	8.0	44%
P1, P2, L3, L4, E2	2	4	1	1	8.0	44%
P4, L1, L4, W2, E1	1	4	2	1	8.0	44%
P4, L1, L2, L6, E2	1	5	1	1	8.0	44%
P7, L1, L2, L3, L4, E2	1	5	0	1	8.0	44%
L1, L2, L3, L4, L5, L6, E1, E2	0	6	0	2	8.0	44%
P6, P7, L4, L5, L6, E1	2	3	0	1	6.0	33%
P4, P8, L1, L2, L4, E1	2	6	2	2	12.0	67%

There are many possible states of the system, each characterized by the loss of different combinations of assets. With the assumptions that assets are either completely present or totally lost, the number of possible states in the configuration in Figure 7-1 is 2^{18} = 262,144. Each state has only one of nineteen values of information infrastructure loss (0.0, 1.0, 2.0, . . ., 18.0), which, when divided by 18.0, gives one of nineteen different percentage information infrastructure losses. The percentage information infrastructure loss for each asset depends on the state of the system immediately prior to the loss of that asset. Note that if the system were in a different state immediately prior to the loss of an asset, then the percent information infrastructure loss associated with the asset's loss might be different. This relationship can be explored systematically. If the procedure of selecting one asset to lose and then recalculating the values of the remaining assets before another one is lost is repeated, always choosing the most valuable asset available gives an upper bound on the information infrastructure loss for any number of assets lost. Always choosing the least valuable asset available gives a lower bound on the information infrastructure loss for any number of assets lost. If the percentage information infrastructure loss is plotted against the number of assets lost, the bounds are as shown in Figure 7-4. The other 262,118 points fall within the area enclosed by the curves. For example, Figure 7-4 also shows the plots for the asset losses shown in Table 7-2. Table 7-2 shows points with 2, 3, 4, 5, 6, and 8 assets lost and 44% information infrastructure lost. Table 7-2 also shows four states with three assets lost that map onto the same point with 44% information infrastructure lost and three states with five assets lost that map onto the same point with 44% information infrastructure lost.

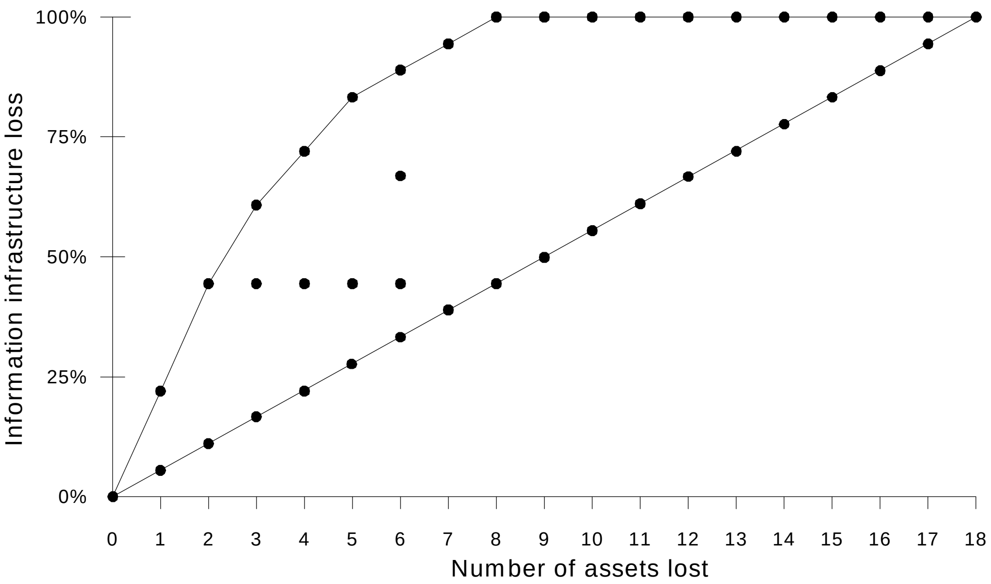


Figure 7-4. Percentage information infrastructure loss vs. number of assets lost (equal-valued assets).

7.2.2 Illustration with different values of assets

In Section 7.2.1, we described a method for determining information infrastructure loss with equal-valued assets. In this section, we extend the analysis to consider the case when assets have different values. For purposes of illustration, assume that the values of the assets in ascending order are: local-area communication, external resource, processor, and wide-area communication. Assume further that the value of an external resource is twice the value of a local-area communication link, the value of a processor is three times the value of a local-area communication link, and the value of a wide-area communication link is four times the value of a local-area communication link. With this information, one may assign a value of 1.0 to a local-area communication link, 2.0 to an external resource, 3.0 to a processor, and 4.0 to a wide-area communication link. Thus, the total information infrastructure value of our Figure 7-1 configuration of 8 processors, 6 local-area communication links, 2 wide-area communication links, and 2 external resources is 42.0. Retaining the assumptions about the relationship between asset loss and information infrastructure loss described in Section 7.2.1, but using the new values, the analysis method proceeds in the same manner, and Table 7-3 shows some of the resulting information infrastructure losses for the loss of different combinations of assets.

Similar to Table 7-2, Table 7-3 focuses attention on combinations of asset losses that result in a 45% (19.0/42.0) loss of information infrastructure. Some combinations are the same in both Tables. The loss of P4 and P8 results in about 45% information infrastructure loss under either valuation. However, most entries are different, with the Table 7-3 usually requiring more assets lost than Table 7-2 to achieve a 45% loss in information infrastructure. The entries in Table 7-3 that show four or five assets lost start with Table 7-2 entries that show three assets lost and add another lost asset. For example, the third row in Table 7-3 shows the three assets from the second row of Table 7-2 lost, P1, P5, and P6. The third row in Table 7-3 adds the loss of E1 to attain a 45% loss of information infrastructure. Similarly, the Table 7-3 entries that show six or seven assets lost start with Table 7-2 entries that show five assets lost. Table 7-3 also shows that as many as eleven assets could be lost from the system with an information infrastructure loss of 45%. If fact, if these assets were lost in a particular order, which is not unique, then each of the

Table 7-3. Information infrastructure loss for asset loss combinations (different-valued assets).

	Information Infrastructure Loss					
Assets Lost	Processors	Local-area	Wide-area	External	Loss	% loss
P4, P8	2	3	2	1	19.0	45%
P2, P8, L4	2	3	2	1	19.0	45%
P2, P4, P5, E1	3	4	1	1	19.0	45%
P1, P5, P6, L6, W2	3	6	1	0	19.0	45%
P4, L1, L4, W2, E1, E2	1	4	2	2	19.0	45%
P1, P2, L3, L4, L6, E1, E2	2	5	1	2	19.0	45%
P1, P7, L1, L2, L3, L4, W1, E2	2	5	1	2	19.0	45%
P1, P5, P6, L1, L2, L3, L4, L5, L6, E1, E2	3	6	0	2	19.0	45%
P6, P7, L4, L5, L6, E1	2	3	0	1	11.0	26%
P4, P8, L1, L2, L4, E1	2	6	2	2	24.0	57%

eleven assets lost would have had value at the time of its loss. Figure 7-5 plots the information infrastructure loss against number of assets lost when assets have the different values stated above. Figure 7-5 shows the plots of the points in Table 7-3 and shows the range of possible relationships between the number of assets lost and percentage information infrastructure loss.

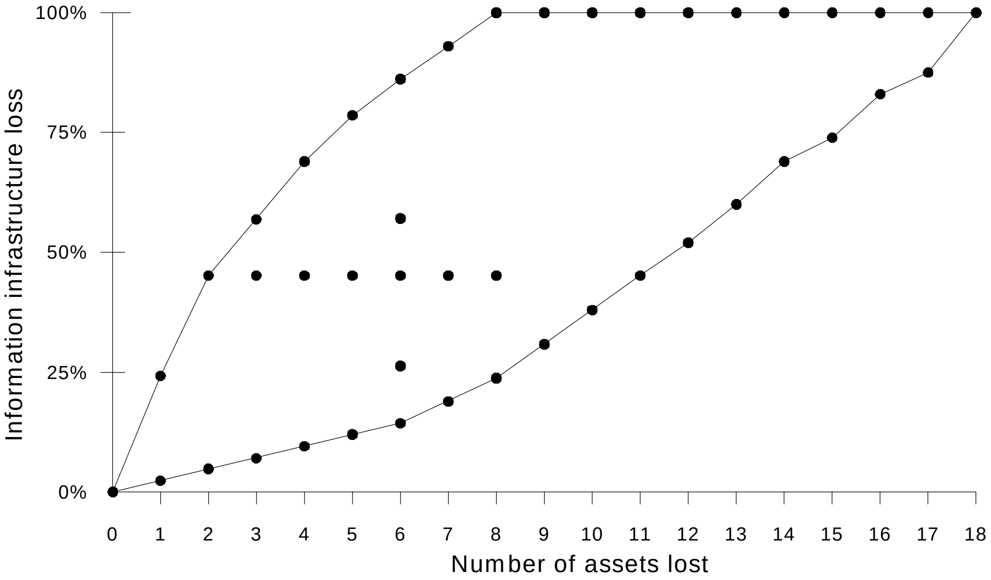


Figure 7-5. Percentage information infrastructure loss vs. number of assets lost (different-valued assets).

7.2.3 Illustration with isolated processor lost

In Section 7.2.1, we described a method for determining information infrastructure loss with equal-valued assets and with the following assumptions with regard to the assets in the system. First that a processor retains its value even if the local- or wide-area communication asset to which it is connected is lost. Second, that a local- or wide-area communication asset has no value if it is connected to only a single processor (i.e., if it is a “dangling wire”). Third, that an external resource has no value unless it is connected to a processor. In this section, we change the analysis to consider the case when a processor is considered lost if it is not connected to another processor by either local or wide-area communications. As with the first illustration, the total information infrastructure value of the Figure 7-1 configuration of 8 processors, 6 local-area communication links, 2 wide-area communication links, and 2 external resources is 18.0. The analysis method proceeds in the same manner, and Table 7-4 shows some of the resulting information infrastructure losses for the loss of different combinations of assets.

Not surprisingly, this illustration shows the possibility of a greater information infrastructure loss with fewer assets lost. Figure 7-6 shows the plots of the points in Table 7-3 and shows the range of possible relationships between the number of assets lost and percentage information infrastructure loss.

Table 7-4. Information infrastructure loss for asset loss combinations (isolated processor considered lost).

	Information Infrastructure Loss					
Assets Lost	Processors	Local-area	Wide-area	External	Loss	% loss
P8	2	1	1	2	6.0	33%
P4, P8	3	3	2	2	10.0	56%
P1, P4, P8	6	5	2	2	15.0	83%
P1, P4, P8, L4	8	6	2	2	18.0	100%
L6, W2	2	1	1	2	6.0	33%
P3, P8	3	2	1	2	8.0	44%
P1, P4	3	4	1	0	8.0	44%
P5, L5, W1, E1	3	3	1	1	8.0	44%
P5, L1, L4, L6, E2	2	4	0	2	8.0	44%
P4, P5, P6, L3, L4, L5, W1, E1	3	3	1	1	8.0	44%
P6, P7, L4, L5, L6, E1	2	3	0	1	6.0	33%
P4, P8, L1, L2, L4, E1	8	6	2	2	18.0	100%

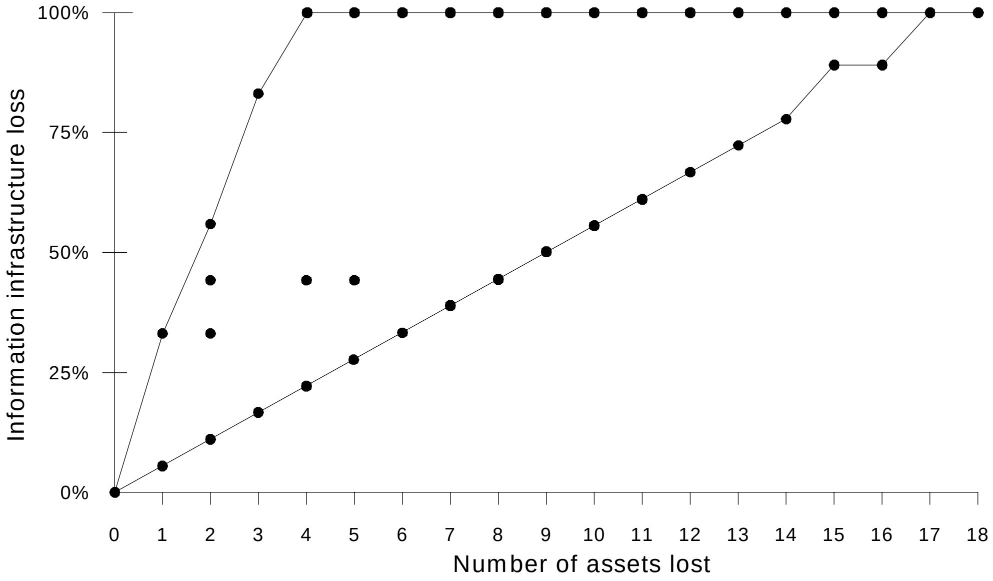


Figure 7-6. Percentage information infrastructure loss vs. number of assets lost (isolated processor considered lost).

8. INFRASTRUCTURE LOSS BASED ON PATH COMPLEXITY

As the nature of the UltraLog system became better defined in the course of the project, the idea arose of basing information system infrastructure on path complexity. That distinct approach was developed and was the version selected for use in the assessments. It is described in this Section and in a Technical Information Report submitted under this contract (Chinnis and Ulvila, 2005).

8.1 Context

To make a survivability claim for UltraLog, it was necessary to demonstrate a particular level of survivability of UltraLog's performance in the face of a given level of information system infrastructure loss. In other work, the UltraLog team is measuring the level of survivability, using experimental data and capability-related utility curves assessed from experts. The other measurement required to make the claim of UltraLog survivability is that of infrastructure. By "infrastructure" is meant the information system (IS) substrate upon which UltraLog and its agents reside. In any case, "infrastructure" is an imprecise term that must be defined here in a way compatible both with common understanding of the term and with quantification.

8.2 Infrastructure Loss—Definition and Measurement

Infrastructure is not the computers, routers, satellite dishes, modems, and such that are used to build a complex dispersed information system. If such equipment were piled in a warehouse, there would be no infrastructure there at all. For that reason, infrastructure cannot be synonymous with the summed cost, value, or information processing potential of the pieces of equipment.

Neither can infrastructure be the functional capability of a *particular* system, such as UltraLog. In fact, the requirement stipulated is that performance be at least 80% when infrastructure loss is 45%. If infrastructure were synonymous with performance, performance necessarily would be 55% when infrastructure loss reached 45%.

So infrastructure is not just the pieces added up and neither is it UltraLog's functionality. What is needed is a way to quantify loss in infrastructure in a complex information system such as the one employed by UltraLog, but not specific to the particular needs and functioning of UltraLog.

Based on earlier work by us in this project (Ulvila, J.W., Boone, J.M., Chinnis, J.O., Jr., & Gaffney, J.E. *Infrastructure loss in UltraLog* (v.1.0). Vienna, VA: Decision Science Associates, Inc., July 2002), and on discussions with various UltraLog project members, several different approaches have been explored. These are briefly characterized in this paper and those that appear to have merit are included in the spreadsheet-based calculators that were used in the project.

8.3 Information System Architecture

The infrastructure associated with the processors, memory, communication links and other hardware depends heavily upon how the parts are arranged and connected. This is largely dictated by the need for functionality of various types at dispersed locations. We will not attempt here to develop a general metric for infrastructure on account of the dependence on architecture dictated by particular system purposes. Instead, we will confine our analysis to the examination of how infrastructure *degrades* in a *particular* architecture: that of the UltraLog test system. The architecture used in the models that are discussed here is the one shown below in Figure 8-1:

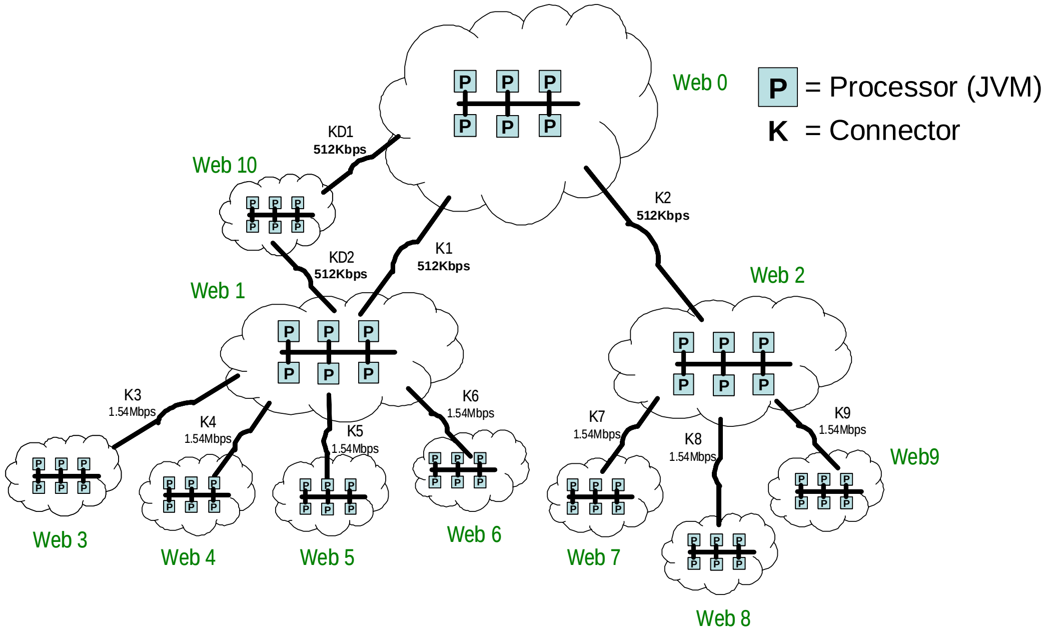


Figure 8-1. Cloud Diagram (Infrastructure Laydown)

Note that “P” represents a processor (CPU and memory), “C” represents its local communications link, processors are arranged in webs, and webs are connected as shown by bi-directional communication links “K” or directed communications “KD.” The KD1 and KD2 links are directed in the sense that they each connect Web 10 with a limited portion of the IS. KD1 connects Web 10 to Web 0 and from there to Webs 2, 7, 8, and 9. KD2 connects Web 10 to Web 1 and from there to Webs 3, 4, 5, and 6. Effectively, Web 10 is not allowed to use the K1 link, and other webs cannot use the KD1 and KD2 links to connect with one another instead of using K1.

Processors themselves are Java virtual machines in the case of the UltraLog test system, and all have equal power, memory, and equivalent C-link connections.

8.4 Methods Considered

Methods explored include the following classes:

Direct judgment of the infrastructure value of IS subunits. This involves assessing the infrastructure loss that occurs, for instance, loss of BW in a K-link as compared with loss of CPU availability. These judgments can be used to construct a model, but they are difficult. In large part the difficulty stems from the tendency to think of infrastructure loss in terms of UltraLog performance when making such judgments. ***Efforts in this area have been successful, however, in leading us to model the infrastructure loss of a node and its C-link as determined by the maximum loss among the CPU, memory, and C-bandwidth.*** Direct assessments closely fit this simple model; further, it is arguable that system designers match CPU, memory, and C to produce maximum performance at a particular cost level, meaning that degradation in one will cause effective collateral loss (decreased functionality) in the others.

Additionally, it was concluded that each K or KD link has no *intrinsic* infrastructure value, instead contributing value to the overall IS by creating paths between processors. It is the combination of CPUs, memory, and C-links into “processors” or nodes and the combination of those processors with K-links and KD-links that constitutes infrastructure.

Modeling of the spread of hierarchical collateral losses. This involved the extension of the ideas developed in explorations of direct judgment to include the ways infrastructure loss would trade off or propagate within an IS. Here, ambiguity arises from the way in which losses might be expected to propagate. Does a loss of processors in a web somehow degrade the infrastructure value of the web above it? If so, how? Does it do so in an additive fashion with losses from its siblings (webs that also connect with the same higher web) or does it do so in some other way? How far does collateral loss extend, and does it progress both up and down the paths in a hierarchy? Are K-links degraded all along a path where a web is degraded, or only to the web above or below? Several possibilities were examined.

These approaches have not been chosen as they emphasized the very strong hierarchical spread of losses and because they involve difficult to support judgments about how the losses spread.

Measurement of infrastructure as a function of path complexity. Perhaps the fundamental way that an IS’s structure is valuable is in the way it allows dispersed resources, consumers, command elements, sensors and effectors to share in the performance of a complex task. For many requirements, a single computer with the same total computing power and memory is worth much less than such a dispersed system because it differs in the *arrangement* of its parts.

The basic principle followed in exploring this approach was the use of the number of paths between processors across the IS as a basis for quantifying infrastructure. In this case, complexity in a fully-connected IS grows in proportion to n(n-1), where n is the number of processors. This approach raises questions of how the infrastructure value of a path (processor-link-processor) is valued with any combination of processor, link, and processor degradations. Possible approaches were explored and one was judged to be a good model for infrastructure loss; it is described briefly below.

Note that in the path-oriented approaches, there are a number of issues to resolve. One is that—while *total* loss of a processor or link eliminates the affected paths from the infrastructure, *partial* processor or link losses (degradations) have a less obvious quantitative effect. For example, should a 50% degradation of all processors across the IS amount to the same degradation loss as a total loss of 50% of the processors (75% infrastructure loss), 50% of the paths (50% infrastructure loss), or something else entirely?

8.5 Description of Selected Path Complexity Model

The capacity loss of a processor is specified as the maximum percentage loss in its CPU (% of GHz lost), memory, M, (% of GB lost), and communications, C, (% of bandwidth lost). The amount of capacity loss in a wide area communications link, K or KD, is its percentage of bandwidth lost (optionally, the % loss below a value judged to be the threshold for losses for that link).

The chosen model of infrastructure loss is a path complexity model based on *specific* processor-to-processor paths. All P-P connections are counted. For a system with no losses, this assigns the same infrastructure value to each processor-to-processor pairing over the entire system. For a fully connected and healthy IS, the total number of paths would be n(n-1)/2, where n is the number of processors.

In this approach, degradations are as follows: The infrastructure contribution of a particular P-P path is proportional to the minimum of the two processor capacities and to the minimum of the capacities of the links that form the path.

As a simple example, consider a fully connected IS with two webs with 8 and 10 processors respectively, for a total n of 18. The infrastructure value of the non-degraded IS is n(n-1)/2, or (18)(17)/2, or 153. Examples of how losses operate are as follows:

- Loss of half the processors leads to an infrastructure value of (9)(8)/2 = 36. This implies an infrastructure loss (proportion) of (153-36)/153 = 0.765. (It can be shown that the infrastructure loss from loss of half the processors ranges from 1.0 when n =2 to 0.75 as n approaches infinity.)
- A uniform degradation of all processors by 50% would lead to an IS with the full number of paths where each path is worth 50% of the original value, or a loss of exactly 0.5. This latter result is independent of n.

The underlying rationale for basing the loss of infrastructure on the loss associated with the processor that is more degraded is that functional capacity of the linked processor pair—in the spirit of needing to honor requirements for interaction-based processing power at specific locations—the path suffers to the degree of the worst degradation. Certainly, it seems reasonable that system with a path operating with both processors at half capacity would have half the capacity of the original path.

Consider the following pairs of processor capacity losses, all of which total 1.0:

- (0, 1) implies a path loss of 1 (total loss);
- (0.5, 0.5) implies a path loss of 0.5; and
- (0.25, 0.75) implies a path loss of 0.75.

In general, for the same total processor capacity loss, infrastructure loss is least where the processor losses are balanced across the two processors and highest when the most unbalanced. This also seems reasonable in terms of the operation of an IS.

The remaining question is how the infrastructure loss associated with a path changes in response to link degradation. The simplest notion is to use the capacity (BW) of the series of links between the two processors as a multiplier on the path's infrastructure value. Infrastructure loss associated with a particular P-P path is therefore proportional to the maximum link loss encountered along the path. Even a degraded processor may still require the full capacity of the link. The size of each link is determined originally by requirements of multiple webs and not by the specific processor pair in question. This is different from the situation with a CPU and its C-link, where a designer has balanced component capacities and the C-link connects specifically to the one processor.

8.6 Calculator

A spreadsheet-based calculator was provided in various forms as an on-going support task for the use of various members of the UltraLog team. The spreadsheet-based calculator was for use by experiment designers, prior to the conduct of experiments. In that way, experiments could be constructed in which different amounts of infrastructure loss were imposed.

The calculators enable calculation of the proportion of infrastructure loss in an IS organized with the web and link structure shown in Figure 8-1. Node labels can be entered for each node in each web, up to 50 nodes in each of the 11 webs. Capacity losses (proportions) can be entered for each node (processor) and for each of the 11 links.

In addition, the methods of the calculator were transitioned to the experiment-related software and implemented there by other UltraLog team members to enable inclusion of infrastructure loss measures in the experimental databases.

8.7 Infrastructure Calculator Examples

Early tests were performed to explore the behavior of the calculator and the underlying infrastructure loss concepts. Mike Dyson of Schafer Corporation devised many of the tests described here. The descriptions refer to the elements shown previously in Figure 8-1.

Scenario description	**Loss**
45% loss in all Ps and Ks	67.1%
K1 cut	42.4%
K2 cut	44.0%
Cut 3 of Ks in lower 1-AD (K3-K5)	36.7%
Cut 3 of Ks in lower 1-AD (K3-K5) plus KD1	44.2%
Cut 2 of lower Ks in 1-UA (K7,K8)	29.8%
All web0 Ps lost	40.7%
All web1 Ps lost	17.8%
All web2 Ps lost	13.2%
Uniform 45% P degradation (includes uniform 45% loss of CPU, memory, and C-links combined)	45.0%
Uniform 30% P degradation (includes uniform 30% loss of CPU, memory, and C-links combined)	30.0%
45% degradation of web0 Ps	18.3%
45% degradation of all Ks	40.3%
Both Web10 links lost	12.7%
Both web1 and web2 Ps lost	29.7%
Loss of approx 45% of Ps in ea web (53 total lost)	69.9%
Loss of nodes (Ps) targeting interesting agent set 1	17.8%
Loss of nodes (Ps) targeting interesting agent set 1 plus mgmt and ca agents in those webs	26.8%
Loss of nodes (Ps) targeting interesting agent set 2	13.2%
Loss of nodes (Ps) targeting interesting agent set 2 plus selected mgmt and ca agents in those webs	19.4%
Loss of nodes (Ps) targeting interesting agent set 3	43.3%
Loss of nodes (Ps) targeting interesting agent set 3 plus selected mgmt and ca agents in those webs	55.4%
Loss of nodes (Ps) targeting interesting agent set 4	44.5%
Loss of nodes (Ps) targeting interesting agent set 4 plus selected mgmt and ca agents in those webs	58.7%

Nodes in each Web:

Web 0	Web 10	Web 1	Web 2
ROOT-CA-NODE	18-MAINTBN-NODE	69-CHEMCO-NODE	NON-CA-CA-NODE
CONUS-NODE	DIVSUP-CSB-NODE	1AD-NODE	NON-CA-MGMT-NODE
NCA-NODE	227-SUPPLYCO-NODE	DIVARTY-1-AD-NODE	UA-BIC-NODE
TRANSCOM-NODE	102-POLSUPPLYCO-NODE	123-MSB-NODE	UA-HHC-NODE
AIR-NODE	592-ORDCO-NODE	1-4-ADABN-NODE	UA-NODE
THEATERGROUND-NODE	106-TCBN-NODE	DISCOM-1-AD-NODE	AVN-DET-NODE
CONUSGROUND-NODE	1-AD-DIVSUP-CA-NODE	141-SIGBN-NODE	NLOS-BN-NODE
SEA-NODE	1-AD-DIVSUP-MGMT-NODE	501-MIBN-CEWI-NODE	UA-FSB-NODE
REAR-A		501-MPCO-NODE	
REAR-B		1-AD-DIV-CA-NODE	
125-ORDBN-NODE		1-AD-DIV-MGMT-NODE	
565-RPRPTCO-NODE			
597-MAINTCO-NODE			
240-SSCO-NODE			
110-POL-SUPPLYCO-NODE			
900-POL-SUPPLYCO-NODE			
REAR-C			
18-PERISH-NODE			
191-ORDBN-NODE			
343-SUPPLYCO-NODE			
REAR-D			
REAR-CA-NODE			
REAR-MGMT-NODE			
CONUS-CA-NODE			
CONUS-MGMT-NODE			
TRANS-CA-NODE			
TRANS-MGMT-NODE			
27	**8**	**11**	**8**

Web 3	Web 4	Web 5	Web 6
1-1-CAVSQDN-NODE	1-BDE-1-AD-NODE	2-BDE-1-AD-NODE	3-BDE-1-AD-NODE
1-501-AVNBN-NODE	501-FSB-NODE	47-FSB-NODE	70-ENG-NODE
127-DASB-NODE	2-37-ARBN-NODE	40-ENGBN-NODE	4-1-FABN-NODE
2-501-AVNBN-NODE	2-3-FABN-NODE	4-27-FABN-NODE	2-70-ARBN-NODE
AVNBDE-1-AD-NODE	16-ENGBN-NODE	2-6-INFBN-NODE	125-FSB-NODE
AVNBDE-NODE	1-37-ARBN-NODE	1-6-INFBN-NODE	1-41-INFB-NODE
AVNBDE-MGMT-NODE	1-36-INFBN-NODE	1-35-ARBN-NODE	1-13-ARBN-NODE
	1-BDE-MGMT-NODE	2-BDE-CA-NODE	3-BDE-CA-NODE
	1-BDE-CA-NODE	2-BDE-MGMT-NODE	3-BDE-MGMT-NODE
7	**9**	**9**	**9**

Web 7	Web 8	Web 9
1-CA-CA-NODE	2-CA-CA-NODE	3-CA-CA-NODE
1-CA-MGMT-NODE	2-CA-MGMT-NODE	3-CA-MGMT-NODE
1-CA-BN-HHC-NODE	2-CA-BN-HHC-NODE	3-CA-BN-HHC-NODE
1-CA-BN-INF-CO-A-NODE	2-CA-BN-INF-CO-A-NODE	3-CA-BN-INF-CO-A-NODE
1-CA-BN-MCS-CO-A-NODE	2-CA-BN-NODE	3-CA-BN-MCS-CO-A-NODE
1-CA-BN-NODE	2-CA-MCS-CO-A-NODE	3-CA-BN-NODE
1-CA-BN-INF-CO-B-NODE	2-CA-BN-INF-CO-B-NODE	3-CA-BN-INF-CO-B-NODE
1-CA-BN-MCS-CO-B-NODE	2-CA-BN-MCS-CO-B-NODE	3-CA-BN-MCS-CO-B-NODE
1-CA-BN-MORTAR-BTY-NODE	2-CA-BN-MORTAR-BTY-NODE	3-CA-BN-MORTAR-BTY-NODE
1-CA-BN-RECON-DET-NODE	2-CA-BN-RECON-DET-NODE	3-CA-BN-RECON-DET-NODE
10	10	10

Interesting Agent Sets

Note: these are the sets of interesting agents, not necessarily the nodes upon which they reside.

Agent Set 1 2nd Brigade plus first tier supply support (11 agents)

2-BDE-1-AD 1-35-ARBN 1-6-INFBN 2-6-INBN

4-27-FABN 40-ENGBN 47-FSB

592-ORDCO 227-SUPPLYCO 102-POL-SUPPLYCO 565-RPRPTCO

Agent Set 1A (17 agents)

Agent Set 1 plus:

2-BDE-CA-NODE 2-BDE MGMT-NODE

DIVSUP-CA-NODE DIVSUP-MGMT-NODE

REAR-CA-NODE REAR-MGMT-NODE

Agent Set 2 CONUS based logistics support (7 agents)

HNS DLA

TRANSCOM GLOBALAIR PLANEPACKER

GLOBALSEA SHIPPACKER

Agent Set 2A

Agent Set 2 plus:

TRANS-CA-NODE TRANS-MGMT-NODE
CONUS-CA-NODE CONUS-MGMT-NODE

Agent Set 3 2 Brigades, Aviation Battalion, and deployed supply chain (29 agents)

1-BDE-1-AD 1-36-INFBN 1-37-ARBN 16-ENGBN
2-3-FABN 2-37-ARBN 501-FSB
2-BDE-1-AD 1-35-ARBN 1-6-INFBN 2-6-INBN
4-27-FABN 40-ENGBN 47-FSB
1-501-AVNBN 127-DASB
123-MSB
592-ORDCO 191-ORDBN 900-POL-SUPPLYCO
227-SUPPLYCO 343-SUPPLYCO 125-ORDBN
102-POL-SUPPLYCO 110-POL-SUPPLYCO 240-SSCO
565-RPRPTCO 597-MAINTCO 18-PERISH-SUBPLT

Agent Set 3A (39 agents)

Agent Set 3 plus:

1-BDE-CA-NODE 1-BDE MGMT-NODE
2-BDE-CA-NODE 2-BDE MGMT-NODE
DIVSUP-CA-NODE DIVSUP-MGMT-NODE
REAR-CA-NODE REAR-MGMT-NODE
DIV-CA-NODE DIV-MGMT-NODE

Agent Set 4 (32 agents)

1-BDE-1-AD 2-37-ARBN 501-FSB
2-BDE-1-AD 1-35-ARBN 1-6-INFBN 2-6-INBN
4-27-FABN 40-ENGBN 47-FSB
1-501-AVNBN 127-DASB
123-MSB
592-ORDCO 191-ORDBN 125-ORDBN

227-SUPPLYCO
343-SUPPLYCO
240-SSCO
102-POL-SUPPLYCO
110-POL-SUPPLYCO
597-MAINTCO
565-RPRPTCO
900-POL-SUPPLYCO
18-PERISH-SUBPLT
HNS
DLA
TRANSCOM
GLOBALAIR
PLANEPACKER
GLOBALSEA
SHIPPACKER

Agent Set 4A (48 agents)

Agent Set 4 plus:

1-BDE-CA-NODE	1-BDE MGMT-NODE
2-BDE-CA-NODE	2-BDE MGMT-NODE
DIVSUP-CA-NODE	DIVSUP-MGMT-NODE
REAR-CA-NODE	REAR-MGMT-NODE
DIV-CA-NODE	DIV-MGMT-NODE
TRANS-CA-NODE	TRANS-MGMT-NODE
CONUS-CA-NODE	CONUS-MGMT-NODE

9. DEGRADATION OF CAPABILITIES AND PERFORMANCE

UltraLog's goal is as follows: "operate with up to 45% information infrastructure loss in a very chaotic environment with not more than 20% capabilities degradation and not more than 30% performance degradation for a period representing 180 days of sustained military operations in a major regional contingency." This section addresses what is meant by the phrases, "20% capabilities degradation" and "30% performance degradation."

UltraLog's capabilities and performance might be considered to be aspects of its functionality. We divide the measures of performance (MOPs) described in the functional assessment, in Section 4 of this report, into those that relate to capabilities and those that relate to performance. These MOPs are then arranged into hierarchies of attributes suitable for multiattribute utility (MAU) analyses. The results of these MAU analyses provide quantitative statements of capabilities and performance that can be used to state percentage degradation in either.

This section demonstrates:

- How the concepts of capabilities and performance are derived from the UltraLog functional assessment;
- How quantitative measures of the capability level and performance level of a system can be determined from MAU analyses;
- How percentage capabilities degradation and percentage performance degradation can be determined from these measures of capability level and performance level.

Fundamentally, UltraLog's capabilities and performance are aspects of its functionality. As shown in Section 4, UltraLog's functional assessment contains the measures of effectiveness (MOEs), operational impacts (OIs) and measures of performance (MOPs) that define functionality. We divide the MOPs described in the functional assessment into those that relate to capabilities and those that relate to performance. These MOPs are arranged into hierarchies of attributes suitable for multiattribute utility (MAU) analyses. The results of these MAU analyses provide quantitative statements of capabilities and performance that can be used to state percentage degradation in either.

9.1 Capabilities and Performance Related to Functional Assessment

For the purpose of this illustration, the MOPs described in the functional assessment in Section 4 can be divided into those that relate to capabilities and those that relate to performance. These MOPs can then be arranged into hierarchies of attributes suitable for multiattribute utility (MAU) analyses. The results of these MAU analyses provide quantitative statements of capabilities and performance that can be used to state the percentage degradation in either. (This description of capabilities and performance is preliminary and serves the purpose of describing the method. Further research may reveal different attributes of capability and performance. In particular, Pigaty (2001) states that all MOPs in the functional assessment are attributes of performance, not capability. If this proves to be the case, then other attributes of capability will need to be developed.)

Capabilities are the things that UltraLog does, performance is how well UltraLog does those things. There is some ambiguity over which functional attributes refer to capabilities and which refer to performance. However, the sets of Interoperability MOPs and User Friendliness MOPs from the second MOE seem to refer primarily to capabilities. The other MOPs refer primarily to performance. If this is so, then a hierarchy of capability attributes could be developed as shown in Figure 9-1, and a hierarchy of performance attributes could be developed as shown in Figure 9-2.

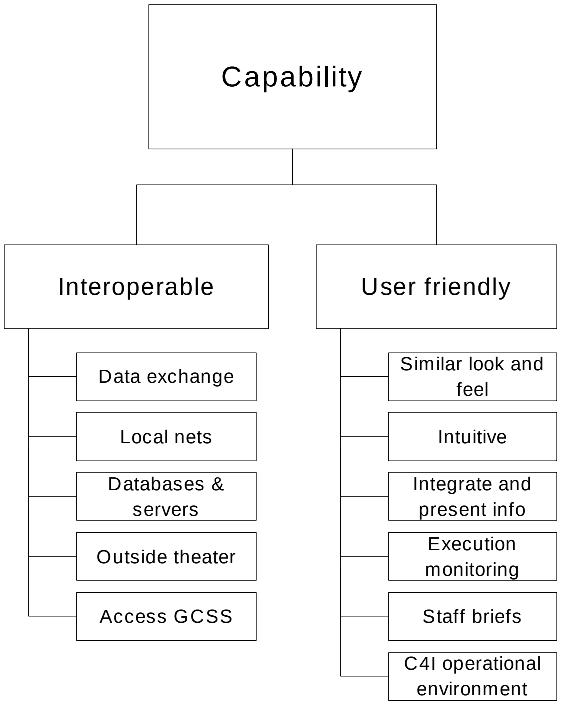


Figure 9-1. MAU hierarchy of capabilities attributes.

The capabilities and performance of any given UltraLog system could be determined from these hierarchies by using multiattribute utility (MAU) methods, as explained in Section 2. This involves the first five steps of MAU analysis: structuring attributes, establishing utility functions, specifying tradeoffs, assessing the performance of systems against attributes, and calculating results. The structure of attributes is the structure shown in Figure 9-1 for capabilities and the structure shown in Figure 9-2 for performance. The second step is to establish a utility function for each attribute. A utility function describes the value or importance that a decision maker assigns to changes in an attribute. In the third step, a set of weights is assessed to describe the tradeoffs among the attributes. Next, the performance of UltraLog is assessed against attributes. This assessment is performed for each condition under which UltraLog capabilities and performance are be evaluated. For example, the assessment would be done for the baseline condition as well as other conditions of interest (e.g., with 45% information infrastructure loss). Finally the results are calculated. One result is a numerical statement of the capability and performance of the system under the conditions assessed. This is on a scale where a utility of 100 represents the value of meeting the capability and performance goals completely, and a utility of 0 represents the bare minimum of acceptability.

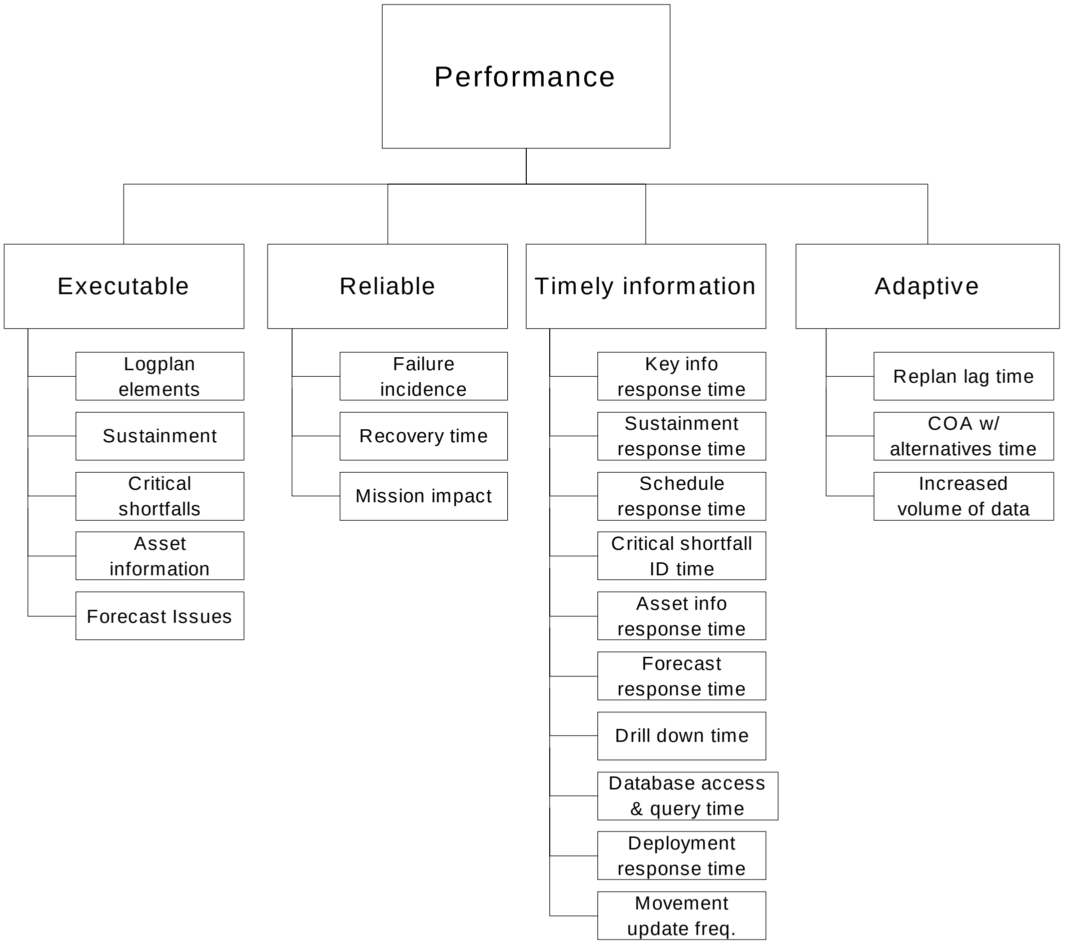


Figure 9-2. MAU hierarchy of performance attributes.

These results present at least two different possible interpretations for the phrases, “20% capabilities degradation” and “30% performance degradation.” One interpretation uses the goals for capabilities and performance as standards. A system with a utility of 80 on capabilities is 80% as valuable as one that met all of the goals for capabilities. Thus, such a system would have a 20% capabilities degradation from the goal. Similarly, a system with a utility of 70 on performance is 70% as valuable as one that met all of the goals for performance. Thus, such a system would have a 30% performance degradation from the goal.

Alternatively, the baseline could be used as the standard. Suppose that the baseline performance of a system was evaluated at a utility of 90 on capabilities and 90 on performance. The baseline capabilities of this system are degraded 10% from the goal and the baseline performance is degraded 10% from the goal. Suppose that, under the condition of a 45% loss in information infrastructure (see Section 6 for a description of a method for determining information infrastructure loss), this same system was evaluated at a utility of 75 on capabilities and 65 on performance. Under this condition, the system is degraded 25% on capabilities and 35% on performance relative to the goal. Using this comparison, the system does not meet the goal of, “operate with up to 45% information infrastructure loss . . . with not more than 20% capabilities degradation and not more than 30% performance degradation.” However, if the baseline system is the basis for comparison, then the 45% loss in information infrastructure produced a degradation of 17% (15 ÷ 90) in capabilities and a degradation of 28% (25 ÷ 90) in performance. The system meets the goal for capabilities and performance degradation relative to the baseline. This latter comparison is the more relevant for assessing the performance of UltraLog.

9.2 Illustration of Performance Degradation

This section contains a complete description of a simplified MAU analysis to determine performance degradation. It illustrates the five steps in an MAU analysis for a reduced set of attributes of performance.

Step 1: Structure performance attributes. Assume, for purposes of this illustration, that the attribute hierarchy for performance is as shown in Figure 9-3. Figure 9-3 contains a subset of the attributes shown in Figure 9-2. The illustration assumes that the attribute, Executable, is described completely by the first and third MOPs, accuracy of key logplan elements and accuracy of the identification of critical shortfalls (see the MOP 1-1-1 and MOP 1-1-4 in Appendix A for a more complete description). Similarly, the attribute, Reliable, is described by the incidence of failure and the time to restore the system after failure (MOP 2-2-1 and MOP 2-2-2). The attribute, Timely Information, is described by the query time for key logplan information and the time to produce level six deployment data (MOP 1-2-1 and MOP 1-2-9). Finally, the attribute, Adaptive, is described by the system lag time for dynamic replanning and the time that it takes the system to develop logistics COA evaluation products with alternatives (MOP 1-3-1 and MOP 1-3-2).

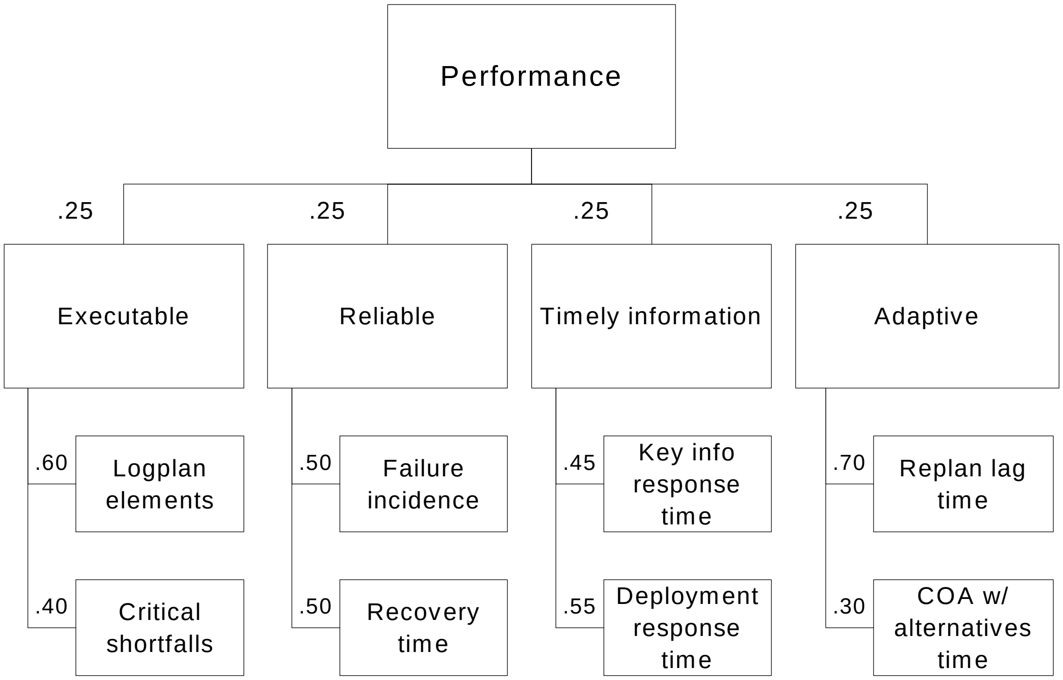


Figure 9-3. Illustrative MAU hierarchy for performance.

Step 2: Establish a utility function for each attribute. The next step in the analysis is to specify a utility function for each attribute that relates the level of performance on the attribute to the value to the decision maker. We can set the utility of the goal performance on the attribute at 100 and the value of a minimally acceptable level of the attribute at 0. Utilities between 0 and 100 show how desirable the performance is as compared with the goal, or, conversely, the level of desirability of the performance relative to the bare minimum. A utility of over 100 is assigned to performance that is better than the goal. Consider the first bottom-level attribute, logplan elements. A metric for this attribute is the percentage error in key logplan elements, one day out. Suppose that the goal is to have an average error of 5%. (Note that all descriptions of utilities and weights used in the examples are hypothetical and for illustration only. They are, however, based on discussions with John Benton and Leo Pigaty of LATA.) Suppose further that the minimum acceptable performance is an average error of 25%. Figure 9-4 shows a possible utility curve for the accuracy of key logplan elements. If the information on all key logplan elements is perfectly accurate (error of 0%), then utility is 120, 20% better than the goal. If the average error is 10%, it is half as good as meeting the goal. Utility is piece-wise linear with the percentage error between these points.

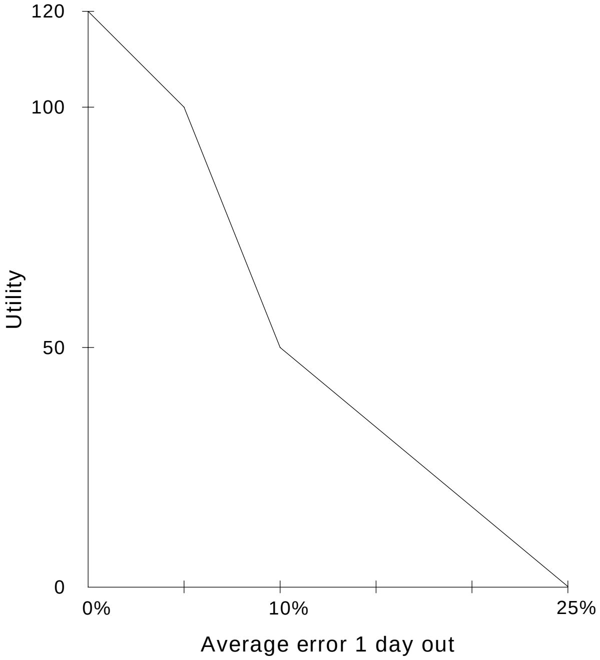


Figure 9-4. Utility function for the accuracy of key logplan elements.

Other shapes may apply for other utility curves, as shown in Figure 4-2. Consider the curves for recovery time and deployment response time. Recovery time shows a utility curve over the time to restore the system to its full operating state after a failure. The curve (curve 2.2.2 in Figure 4-2) shows the utility curve for the .80 fractile of recoveries. That is, the time necessary for recovery from 80% of the failures. This fractile was chosen instead of the mean recovery time to better reflect the more extreme part of the distribution of recovery times. The goal is for the system to be restored to its full operating state within 5 minutes 80% of the times that it fails. Fifteen minutes is too long. The utility is linear with time between 5 and 15 minutes. A utility of 120 is assigned to instantaneous recovery.

The utility curve for the time that it takes the system to produce level six deployment data for a contingency (curve 1.2.9 in Figure 4-2) is S-shaped between a maximum utility of 135 at 0 minutes and a minimum of 0 utility at 120 minutes. A utility of 100 for 60 minutes represents the goal of, “level six deployment data for a contingency in one hour” from the “UltraLog Active Assessments Plan” by Saydjari, Wood, and Bouchard (2001).

Utility functions can also be displayed in tabular form. Table 9-1 shows the utility functions for all of the attributes in the example. In all cases shown, utility is assumed to be piece-wise linear with the metric shown between the values in the table.

Table 9-1. Utility functions for all performance attributes in the example.

Logplan elements	% error 1 day out	0%	5%	10%	25%			
	utility	120	100	50	0			
Critical shortfalls	% identified	75%	85%	90%	100%			
	utility	0	50	100	120			
Failure incidence	failures/day	0	2	3	4	5		
	utility	120	100	50	20	0		
Recovery time	minutes (.80 fractile)	0	5	15				
	utility	120	100	0				
Key info response time	minutes (.95 fractile)	0	0.5	1	2	3		
	utility	140	100	60	20	0		
Deployment resp. time	minutes	0	30	60	75	90	105	120
	utility	135	125	100	75	25	10	0
Replan lag time	minutes	0	15	30				
	utility	125	100	0				
COA with alternatives	minutes	0	15	30				
	utility	125	100	0				

Step 3: Establish tradeoffs (weights). The next step is to establish tradeoffs across attributes. In an additive MAU analysis, tradeoffs are expressed as weights. The weights are used to compare utilities across attributes and to combine the utilities on single attributes into utilities on groups of attributes. When assessing weights, one compares the importance of the swing on one attribute scale with the swing on other attribute scales.

For our example, assume that all swings being compared are those between 0 and 100 utilities on the attributes. Starting with the subattributes under the attribute, Executable, assume that it is 50% more valuable to reduce the error in logplan elements from 25% to 5% than it is to improve the percent of critical shortfalls identified from 75% to 90%. This assessment is consistent with normalized swing weights of 0.60 for logplan elements and 0.40 for critical shortfalls. Next, assume that it is as valuable to reduce major database access failures from 5 per day to 2 per day as it is to reduce the 0.80 fractile recovery time from 15 minutes to 5 minutes. This gives normalized swing weights of 0.50 to both failure incidence and recovery time. If the improvement from 120 minutes to 60 minutes on deployment response time is a little more valuable than improving from 3 minutes to 30 seconds on key information query response time, then key information response time might have a normalized swing weight of 0.45, and deployment response time might have a normalized swing weight of 0.55. Similarly, replan lag time might have a normalized swing weight of 0.70, and course of action with alternatives might have a normalized swing weight of 0.30. These weights are used in the rest of the example.

The final set of weights needed for a performance evaluation is one across all of the attributes of performance. Assume that the improvement from the worst to the goal level of each of these attributes is of equal value. This gives normalized swing weights of 0.25 for each of the four attributes, Executable, Reliable, Timely Information, and Adaptive. Normalized swing weights are shown in Figure 9-3.

Step 4: Assess performance against attributes. The next step in our example is to specify the performance of the system under evaluation against each of the attributes. This example illustrates the method with one system, but with an evaluation of its baseline performance (here baseline refers to the system with full information infrastructure) and its performance under the condition of 45% information infrastructure loss. (See Section 6 for a method to determine information infrastructure loss.) The analysis also shows assessments for two hypothetical systems, one that meets the goal on every attribute, and one that is at the minimum level on every attribute. Table 9-2 shows an example of the assessment.

Table 9-2. Assessments of the performance of systems against attributes.

Attributes	Baseline	45% infrastructure loss	Minimum	Goal
Logplan elements	7% error	10% error	25% error	5% error
Critical shortfalls	85% identified	83% identified	75% identified	90% identified
Failure incidence	2.5 failures/day	3 failures/day	5 failures/day	2 failures/day
Recovery time	8 minutes	9 minutes	15 minutes	5 minutes
Key info response	45 seconds	1 minute	3 minutes	30 seconds
Deployment response	60 minutes	75 minutes	120 minutes	60 minutes
Replan lag time	15 minutes	21 minutes	30 minutes	15 minutes
COA w/ alternatives	20 minutes	20 minutes	30 minutes	15 minutes

Step 5: Calculate results. The final step is to calculate the performance utility of the systems. A weighted average utility is calculated at each level in the hierarchy using the normalized weights from Step 3 and the utility functions from Step 2 applied to the assessments in Step 4. Table 9-3 shows results at all levels in the hierarchy.

Consider the column for Baseline. The numbers shown in a right alignment are the utilities for the performance levels shown in Table 9-2. For example an average error of 7% in key logplan elements 1 day out has a utility of 80, correct identification of 85% of critical logistics shortfalls has a utility of 50, 2.5 failures to access major databases per day has a utility of 75, a failure recovery time of 8 minutes from 80% of failures has a utility of 70, and so forth (all utilities are rounded to the nearest whole number throughout the example).

Table 9-3. Calculation of performance.

	weight	Baseline	45% infrastruc-ture loss	Minimum	Goal
PERFORMANCE		**80**	**58**	**0**	**100**
Executable	0.25	68	46	0	100
Logplan elements	0.60	80	50	0	100
Critical shortfalls	0.40	50	40	0	100
Reliable	0.25	72	55	0	100
Failure incidence	0.50	75	50	0	100
Recovery time	0.50	70	60	0	100
Timely information	0.25	91	68	0	100
Key info response	0.45	80	60	0	100
Deployment response	0.55	100	75	0	100
Adaptive	0.25	90	62	0	100
Replan lag time	0.70	100	60	0	100
COA w/ alternatives	0.30	67	67	0	100

The utilities at the intermediate level in the hierarchy of attributes are calculated as weighted averages of the utilities of their subattributes. For example, the utility of the Baseline system on the attribute, Executable, is calculated as: (0.60)(80) + (0.40)(50) = 68. These utilities are then used to calculate the utilities for overall performance, but now the weights are those of the intermediate attributes. For example, the performance utility of the Baseline is calculated as:

$$(0.25)(68) + (0.25)(72) + (0.25)(91) + (0.25)(90) = 80.$$

These calculations show that the Baseline system has a performance degradation from the goal of 20%. Under the condition of 45% information infrastructure loss, the system is degraded by 42% in performance from the goal. If the goal performance is used as the standard for determining whether a system meets the goal of: “operate with up to 45% information infrastructure loss . . . with . . . not more than 30% performance degradation,” then the system fails to meet this goal. However, if the baseline performance is used as the standard of comparison, then the system meets the goal. With 45% information infrastructure loss, the system’s performance is degraded by 28% from the baseline performance (22 ÷ 80).

9.3 Illustration of Capabilities Degradation

The method for determining capabilities degradation is the same as the method described in Section 9.2 for determining performance degradation. This section repeats a similar analysis for capabilities.

Step 1: Structure attributes. The structure for capabilities attributes for the example is shown in Figure 9-5. The attributes shown are a subset of those in Figure 9-1.

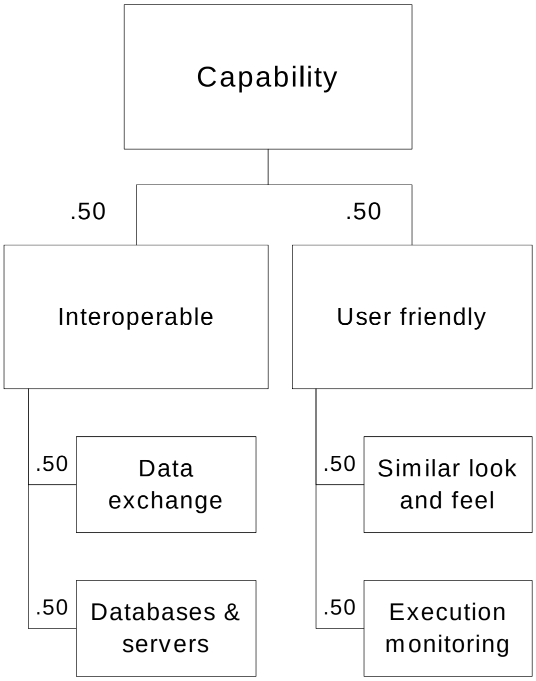


Figure 9-5. Illustrative MAU hierarchy for capabilities.

Step 2: Establish a utility function for each attribute. The utility functions for the capabilities example are shown in Table 9-4. This set has one utility function for an attribute, Similar Look and Feel, that does not have a corresponding metric. Instead, Similar Look and Feel is characterized by one of one of five qualitative descriptors. This illustrates that the method does not depend on having metrics for all attributes in order to determine a numerical utility.

Table 9-4. Utility functions for all capabilities attributes in the example.

Data exchange	% real-time, secure	80%	85%	90%	95%	100%
	utility	0	10	50	100	120

Databases & servers	%	95%	100%			
	utility	0	100			
Similar look & feel familiar	qualitative descriptor	Few	Some	Many	Most	All
	utility	0	25	50	75	100
Execution monitoring	% of items	70%	90%	100%		
	utility	0	100	120		

Step 3: Establish tradeoffs (weights). Normalized swing weights are shown for all attributes in Figure 9-5 In this example, each attribute has the same swing weight.

Step 4: Assess performance against attributes. Table 9-5 shows illustrative assessments against the capabilities attributes for the same systems used in the performance example.

Table 9-5. Assessments of the performance of systems against capabilities attributes.

Attribute	Baseline	45% infrastructure loss	Minimum	Goal
Secure communication	94% real-time, secure	92% real-time, secure	80% real-time, secure	95% real-time, secure
RDBMS interoperable	Interoperable with 99.5%	Interoperable with 98.5%	Interoperable with 95%	Interoperable with 100%
Look and feel familiar	Most familiar	Most familiar	Few familiar	All familiar
Execution monitoring	89% of items	84% of items	70% of items	90% of items

Step 5: Calculate results. Table 9-6 summarizes the calculations for capabilities.

Table 9-6. Calculations for capabilities.

	weight	Baseline	45% infrastruc-ture loss	Minimum	Goal
CAPABILITY		**88**	**71**	**0**	**100**
Interoperable	0.50	90	70	0	100
Secure communication	0.50	90	70	0	100
RDBMS interoperable	0.50	90	70	0	100
User Friendly	0.50	85	73	0	100
Look and feel familiar	0.50	75	75	0	100
Execution monitoring	0.50	95	70	0	100

These calculations show a result similar to the one shown for performance. The Baseline system has a capabilities degradation from the goal of 12%. Under the condition of 45% information infrastructure loss, the system is degraded by 29% in capabilities from the goal. If the goal capability is used as the standard for determining whether a system meets the goal of: “operate with up to 45% information infrastructure loss . . . with not more than 20% capabilities degradation,” then the system fails to meet this goal. However, if the baseline performance is used as the standard of comparison, then the system meets the goal. With 45% information infrastructure loss, the system’s performance is degraded by 19% from the baseline performance (17 ÷ 88).

10. ROBUSTNESS

The methods described in this report can be used to provide a practical way to assess the robustness of an UltraLog system. We propose that, for UltraLog, *robustness* might be defined as the property of UltraLog of minimizing the effects of the causes of variation without eliminating the causes (See Appendix C of Ulvila *et al*., 2001 for a detailed discussion). A particular interest is when information infrastructure loss due to kinetic or information warfare attacks causes the variation. Consistent with the UltraLog goal, one could be interested in the effects of any level of information infrastructure loss from 1% to 45%. However, in practice it is likely that the specification of a few particular levels of information infrastructure loss will provide a reasonable approximation to the full range of interest.

Robustness might also refer to the performance of UltraLog under conditions of chaos in the environment. Currently, there is no agreed definition of this environmental chaos, yet the UltraLog goal is to, "operate with up to 45% information infrastructure loss in a very chaotic environment with not more than 20% capabilities degradation and not more than 30% performance degradation for a period representing 180 days of sustained military operations in a major regional contingency," (UltraLog website). The dimensions of chaos in the environment have been debated by the Metrics and Assessment Working Group, as summarized by Ulvila (2001). It has been suggested that chaos might include: severe imbalances in the logistics system, local distractions to logisticians, disruptions in the external infrastructure, aberrant operator behavior, loss of confidence by logisticians in the UltraLog system, and information infrastructure loss. Pigaty (2001) analyzed these suggested dimensions of chaos and concluded that environmental chaos could be described in terms of OPTEMPO and availability of communications. High OPTEMPO combined with low availability of communication is very chaotic (level 4 chaos), followed by low OPTEMPO with low communications availability (level 3 chaos), high OPTEMPO with high communications availability (level 2 chaos), and finally, low environmental chaos for low OPTEMPO with high communications availability (level 1 chaos). About the only area of agreement is that chaos in the environment is a type of disorder that could adversely affect the operation of UltraLog. However it is eventually resolved, chaos, along with information infrastructure loss, is important to the operational definition of robustness.

Figure 10-1 shows a proposed MAU structure for determining robustness. The goal of UltraLog is operation "with up to 45% information infrastructure loss in a very chaotic environment," so this point is shown as the most severe case for the assessment. The point of 25% information infrastructure loss in a moderately chaotic environment is shown as another evaluation point. If necessary, other causes of variation could be included, but, at this time, it appears that information infrastructure loss and chaos can represent adequately the important causes of variation of performance.

Section 9 of this report illustrates how one might determine quantitative measures of the capabilities and performance of an UltraLog system. The outputs from these analyses, the percent degradation in capabilities and the percent degradation in performance can be used as inputs to an MAU analysis to assess the robustness of the system under evaluation. Repeating this analysis for subsequent versions of UltraLog shows how progress is being made toward the goal. The following is an example.

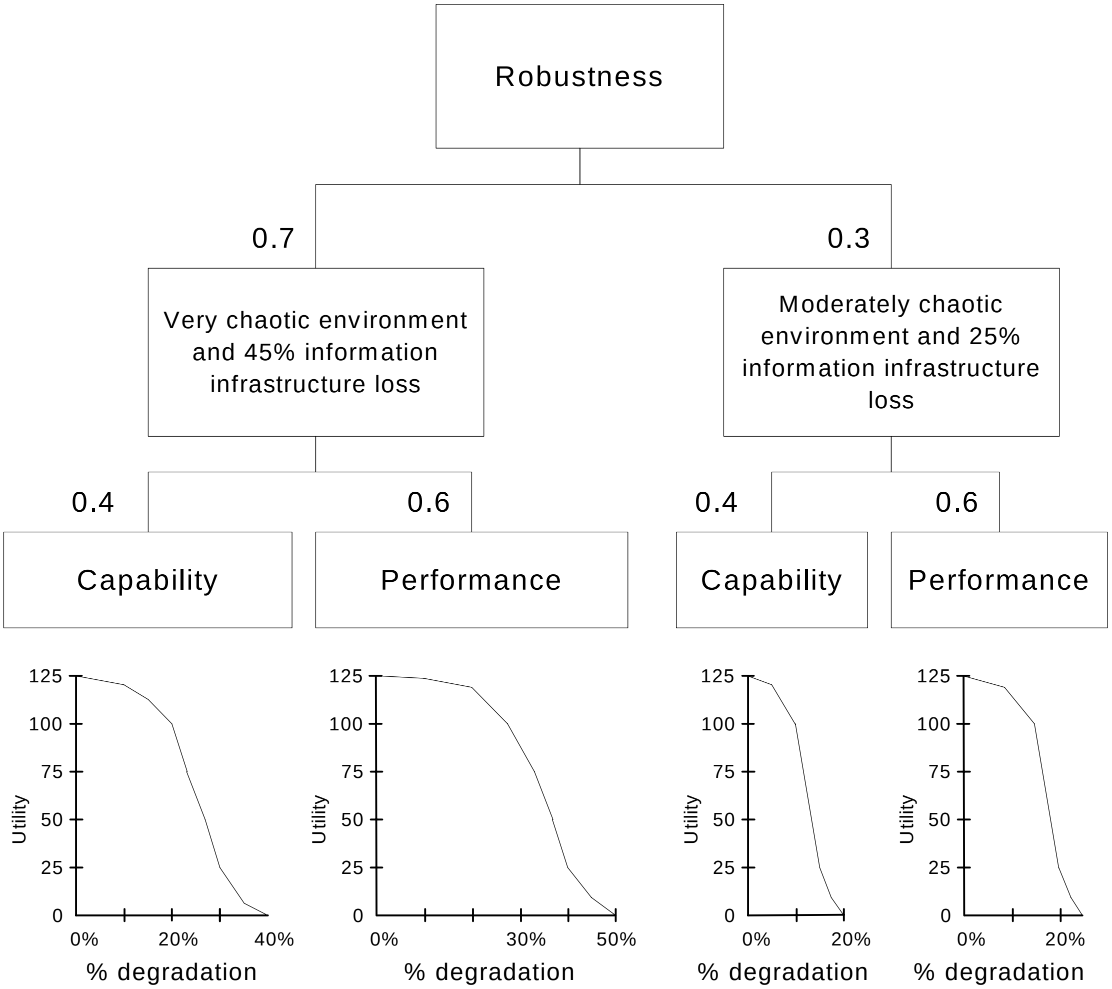

Robustness
0.7
0.3
Very chaotic environment and 45% information infrastructure loss
Moderately chaotic environment and 25% information infrastructure loss
0.4
0.6
0.4
0.6
Capability
Performance
Capability
Performance
125
100
75
50
25
0
Utility
0%
20%
40%
30%
50%
% degradation

Figure 10-1. MAU structure for robustness.

Step 1: Structure attributes. The structure for robustness attributes is shown in Figure 8-1. Robustness is determined from the percentage degradation in capabilities and performance under two conditions: in a very chaotic environment with 45% information infrastructure loss and in a moderately chaotic environment with 25% information infrastructure loss.

Step 2: Establish a utility function for each attribute. Examples of utility functions for the robustness attributes are shown in Figure 8-1. In each case, utility is shown as a reflected S-shaped function of the attribute's metric, percent degradation. A utility of 100 is assigned to a value that fully meets the goal. In the very chaotic environment with 45% information infrastructure loss, the goals are 20% capabilities degradation and 30% performance degradation, which are taken from UltraLog's goal. Smaller degradations are better than the goal, and the utility function continues to increase up to a utility of 125 with no degradation. For this illustration, minimum acceptable levels of 40% capabilities degradation and 50% performance degradation receive utilities of 0. In between the goal level and the minimum level, utility drops sharply to 25 and then more slowly as the minimum level is approached. In the example, the utility goals and minimum acceptable levels in a moderately chaotic environment with 25% information infrastructure loss are assumed to be half the corresponding levels for a very chaotic environment with 45% information infrastructure loss. Points on all four utility functions are tabulated in Table 10-1.

Table 10-1. Utility functions for robustness attributes.

Capability @ 45%	degradation	0%	10%	15%	20%	23%	27%	30%	35%	40%
	utility	125	120	115	100	75	50	25	10	0
Performance @ 45%	degradation	0%	10%	20%	30%	33%	37%	40%	45%	50%
	utility	125	123	119	100	75	50	25	10	0
Capability @ 25%	degradation	0%	5%	7.5%	10%	12%	13%	15%	17%	20%
	utility	125	120	115	100	75	50	25	10	0
Performance @ 25%	degradation	0%	5%	10%	15%	17%	18%	20%	22%	25%
	utility	125	123	119	100	75	50	25	10	0

Step 3: Establish tradeoffs (weights). For this example, assume that meeting the goals for capabilities and performance for a very chaotic environment with 45% information infrastructure loss is a little more than twice as important as meeting the goals for a moderately chaotic environment with 25% information infrastructure loss. Normalized swing weights of 0.7 for 45% information infrastructure loss and 0.3 for 25% information infrastructure are consistent with this assumption. Within each condition, assume that it is 50% more important to meet the performance goal (i.e., the swing from the minimum level to the goal level) than it is to meet the capabilities goal. This judgment gives a normalized swing weight of 0.6 to performance and 0.4 to capability. These are the swing weights shown in Figure 10-1.

Step 4: Assess performance against attributes. For this analysis, the inputs, degradations in capabilities and performance, would come from analyses such as those described in Section 9. That is, MAU analyses would be conducted for the systems using the capability attributes shown in Figure 9-1 and the performance attributes shown in Figure 9-2 to evaluate the performance of the system under both conditions. The results of these analyses would be compared with either the goal capabilities and performance or with the baseline capabilities and performance to give capabilities degradation and performance degradation. Those are the inputs to the robustness evaluation. Table 10-2 shows illustrative results from analyses to determine capabilities and performance degradations of UltraLog over time (labeled UltraLog 01, UltraLog 02, and UltraLog 03) as inputs to an analysis of robustness.

Table 10-2. Assessments for UltraLog systems against robustness attributes.

Attributes	UltraLog 01	UltraLog 02	UltraLog 03	Goal
Capability @ 45%	32.0%	25.0%	19.0%	20.0%
Performance @ 45%	45.0%	36.0%	31.0%	30.0%
Capability @ 25%	15.0%	13.0%	11.0%	10.0%
Performance @ 25%	20.0%	17.5%	13.0%	15.0%

Step 5: Calculate results. Table 10-3 summarizes the calculations for robustness. This analysis contributes the robustness portion of a complete analysis of functionality, cost, and survivability. The following paragraphs illustrate some of the uses of this robustness result.

Table 10-3. Robustness calculation.

	weight	UltraLog 01	UltraLog 02	UltraLog 03	Goal
ROBUSTNESS		**17**	**59**	**97**	**100**
@ 45% Infrastructure loss	0.7	14	58	97	100
Capability @ 45%	0.4	19	63	103	100
Performance @ 45%	0.6	10	55	92	100
@ 25% Infrastructure loss	0.3	25	60	98	100
Capability @ 25%	0.4	25	55	85	100
Performance @ 25%	0.6	25	63	108	100

What is the robustness of the current UltraLog development? This is an evaluation that has interest in its own right and could contribute to the larger evaluation of UltraLog functionality and survivability. Table 10-3 shows, for example, that UltraLog 01 has an evaluation of 17. This evaluation is interpretable as a level of robustness on a scale where 0 corresponds to a situation in which all aspects of robustness are at the minimum acceptable levels, and 100 corresponds to a situation in which all aspects of robustness are at the goal levels. The utility functions were constructed to be cardinal scales so the evaluation numbers carry more information than a simple ordering, and this information is useful for answering the next questions.

How does this compare with the requirements and goals? The goal represents a hypothetical system that contains the goal level of every robustness attribute. It has a utility of 100 for each attribute, and its overall utility is 100. The robustness evaluations of the systems can be compared with the goal. Table 10-3 shows that UltraLog 01 falls well short of the goal for all robustness attributes. Its overall evaluation of 17 indicates that it is 17% of the way from a minimally acceptable system to the goal. Its best performance is only 25% of the goal.

How are the evaluations changing over time? If UltraLog 01, UltraLog 02, and UltraLog 03 are subsequent versions of UltraLog, then an examination of their evaluations shows how evaluations are changing over time. UltraLog 01 is evaluated at 17% of the goal for robustness, UltraLog 02 at 59%, and UltraLog 03 at 97%. Figure 10-2 displays this information graphically and shows a steady progress toward the overall goal, but never quite reaching the goal for robustness.

What attributes need to be enhanced to improve the robustness evaluation? Any attribute with a utility of less than the maximum could be improved to increase the robustness evaluation. The analysis shows that there is much room for improvement in UltraLog 01 and UltraLog 02. UltraLog 01 is low on all attributes of robustness. UltraLog 02 is at a middle level on all attributes of robustness. Figure 10-3 shows how the utilities of the systems for the robustness attributes can be displayed graphically to show the room for improvement and changes over time. This display also highlights the fact that UltraLog 03 falls significantly short of the goal on capability at 25% loss of information infrastructure but is better than the goal on both capability at 45% loss of information infrastructure and performance at 25% loss of information infrastructure. When considering the relative importance of achieving the goals, the net is that UltraLog 03 is a little short of the goal for robustness.

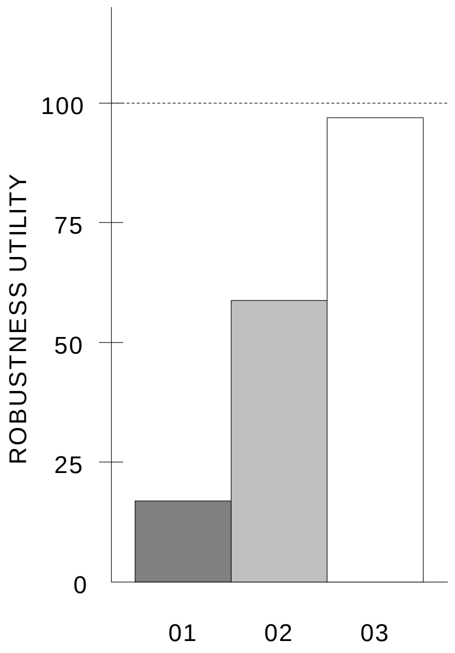


Figure 10-2. Robustness evaluations.

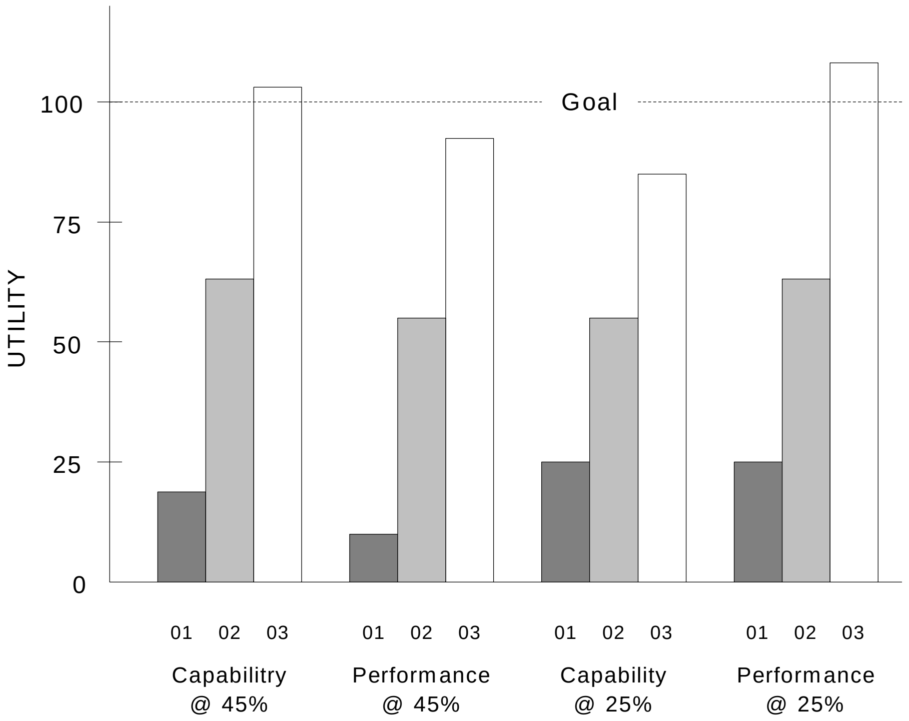


Figure 10-3. Evaluations against attributes of robustness.

11. MAUL: A PROTOTYPE SOFTWARE TOOL

This section describes the prototype software tool, MAUL (MultiAttribute utility for UltraLog). MAUL is a proof-of-concept prototype that demonstrated the feasibility of using multiattribute utility analyses in UltraLog. The methods and algorithms were then incorporated into other UltraLog software by S/TDC, where it was employed to produce the experiment databases.

The prototype was delivered separately as a set of computer files, each containing a different analysis that is described in this report. Files were delivered that contained the functional assessment described in Section 4, the Red Team security assessment described in Section 5, the illustration of performance degradation described in Section 7.2, the illustration of capabilities degradation described in Section 7.3, and the analysis of robustness described in Section 8. All illustrations used in Section 11 are taken from the functional assessment.

This section describes the prototype and its use. It illustrates the input, output, and analysis options of MAUL and how to use them.

11.1 Functionality

MAUL supports all aspects of the multiattribute utility analysis for UltraLog that are described in this report. This includes the specification of a hierarchical structure of multiple attributes, the specification of a utility curve for each attribute, the specification of weights, the specification of the performance of systems or designs against the attributes, calculation of results, and performance of sensitivity analyses. The four most important sensitivity analyses for MAU analyses, which are described in Section 2.4, are supported by MAUL.

11.2 Operating Environment

MAUL operates under Microsoft Excel. It was developed using Excel version 9.0 (Microsoft Excel 2000). MAUL should operate without modification on Macintosh systems, provided that Excel v9.0 is present, but it has not been tested in that environment.

11.3 Description

Because our prototype MAUL is a spreadsheet, certain spreadsheet operations can be used, such as formatting cells, setting print ranges, and copying data into input ranges. Hiding and unhiding selected columns and rows can also be used to simplify and restore complex displays. MAUL has been tested to verify that it performs the calculations of a multiattribute utility analysis correctly. However, MAUL is a prototype that demonstrates the feasibility of developing a decision support tool, it is not the finished tool. In particular, MAUL includes few features that guard against misuse. Operations that change the structure of the sheets will produce errors and cause the formulas to fail. Such operations include inserting, moving, or deleting worksheets, rows, columns, or cells.

The worksheets are protected (protection is turned on) in order to prevent accidental overwriting of formulas or other changes to the tool. No password is used. To change a cell’s format, hide a column, or make any legal change to a worksheet it is necessary to first turn off protection (Tools/Protection/UnprotectSheet on the Excel menu bar), make the desired change, and then turn protection back on.

If a failure occurs, the analysis should be reconstructed from the point before the failure. We recommend that the delivered MAUL files be retained as working archival versions of MAUL. Any modifications should be saved under new names. In this way, the user will be assured of having an uncorrupted version of MAUL in the archives.

11.3.1 Basic functional organization

MAUL is presently implemented as an Excel spreadsheet with seven basic sheets normally accessible via their tabs. These are as follows:

- Input Structure: An active form for entering the hierarchy of attributes, utility function for attributes, weights across attributes at all levels in the hierarchy, and the names and assessments for the systems under evaluation.
- Utility Curve: Tabular and graphical displays of the utility curve for a specified attribute.
- Show Node: A tabular display of the MAU analysis at any specified node in the hierarchy. It also contains a complete list of all nodes in the hierarchy.
- Discrimination: A tabular display of the sensitivity analysis that compares the utilities of two systems.
- Cum. Wt. Sort: A Tabular list of the attributes in order of their cumulative weights in the analysis.
- Local Wt. Sensitivity: A graphical display of the sensitivity of the overall evaluation of systems to changes in the local weight given to a specified attribute.
- Cum. Wt. Sensitivity: A graphical display of the sensitivity of the evaluation of systems to changes in the cumulative weight given to a specified attribute.

11.3.2 Input structure

The input structure tab is an active sheet for entering the basic inputs to an MAU analysis: the hierarchy of attributes, utility function for attributes, weights across attributes at all levels in the hierarchy, and the names and assessments for the systems under evaluation.

Enter the hierarchy of attributes in columns A through G and I, rows 5 through 54. Enter the names of the attributes in column I. Enter the locations of the attributes in the hierarchy in columns A through G. The top attribute in the hierarchy is assumed to be “node zero,” and it does not need an identifying number. The rest of the hierarchy is numbered in the same manner as the outline used in the functional assessment in Appendix A. Table 11-1 shows an example for the first part of the functional assessment. Column A has the designator for the first level below the top. A value of 1 indicates that the attribute is part of the first MOE. The number in Column B indicates the OI, and the number in Column C indicates the MOP. If the hierarchy had more levels, then additional columns (D-G) would have been used. Enter the entire structure and then leave the rest of the rows blank. MAUL will automatically know when to stop.

Enter information about the utility functions for the attributes in columns J, AR through BB, and BC through BM. Column J designates whether the attribute has a metric. Enter TRUE if the attribute has a metric and leave the cell blank if not. If the attribute has a metric, enter up to 11 values (labeled “Level”) of the metric in columns AR through BB, and enter the corresponding utilities in columns BC through BM. Entries must be made in *increasing* value of the *metric*. The corresponding utility could be increasing, decreasing, or non-monotonic. Be sure to cover the full range of the utility function. The tool will not extrapolate beyond the entries. It will highlight entries that fall below the lowest entry. Utilities must be greater than or equal to zero. Examples of some of the utility functions are shown in Table 11-2. These correspond to some of the utility curves shown in Figure 4-2.

Table 11-1. Partial example of the functional assessment hierarchy of attributes.

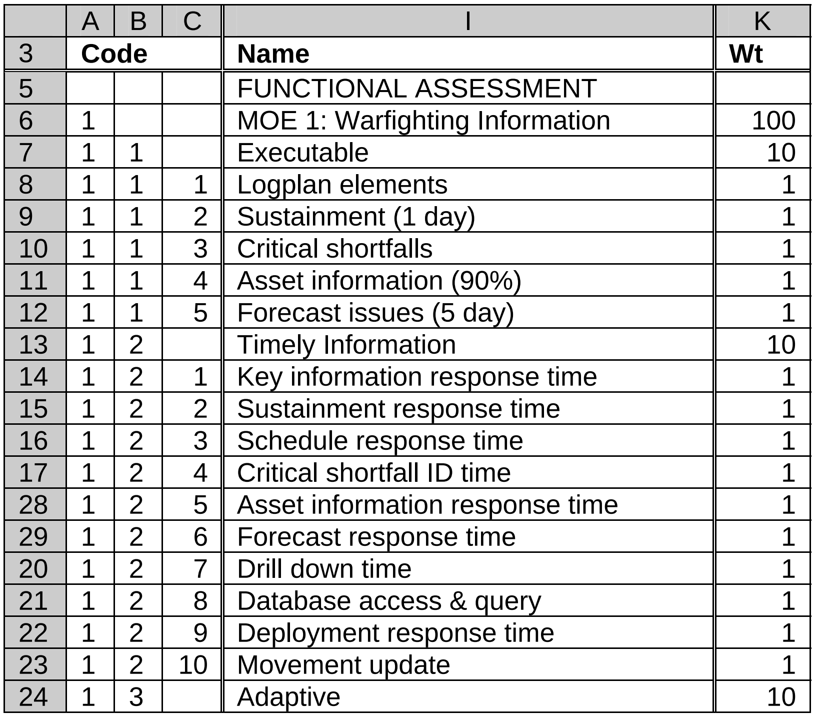
	A	B	C	D	E	F	G	I
3	**Code**							**Name**
5								FUNCTIONAL ASSESSMENT
6	1							MOE 1: Warfighting Information
7	1	1						Executable
8	1	1	1					Logplan elements
9	1	1	2					Sustainment (1 day)
10	1	1	3					Critical shortfalls
11	1	1	4					Asset information (90%)
12	1	1	5					Forecast issues (5 day)
13	1	2						Timely Information
14	1	2	1					Key information response time
15	1	2	2					Sustainment response time
16	1	2	3					Schedule response time
17	1	2	4					Critical shortfall ID time
28	1	2	5					Asset information response time
29	1	2	6					Forecast response time
20	1	2	7					Drill down time
21	1	2	8					Database access & query
22	1	2	9					Deployment response time
23	1	2	10					Movement update
24	1	3						Adaptive

Table 11-2. Examples of utility functions.

	A-C	J	AR	AS	AT	AU	BC	BD	BE	BF
3	Code	Metric	Level 1	Level 2	Level 3	Level 4	Utility 1	Utility 2	Utility 3	Utility 4
5	1 1 1	TRUE	0%	5%	10%	25%	120	100	50	0
10	1 1 3	TRUE	75%	85%	90%	100%	0	50	100	120
23	1 2 10	TRUE	0	4	100		120	40	0	
25	1 3 1	TRUE	0	15	30		125	100	0	
27	1 3 3	TRUE	50%	60%	80%	100%	0	60	100	115
31	2 1 2	TRUE	95%	100%			0	100		
37	2 2 2	TRUE	0	5	15		120	100	0	

Enter relative local swing weights for the attributes in Column K, rows 5 through 54. Weights are relative in that they do not have to add to 1.0, the tool automatically normalizes the weights. For example, if the weights are all the same, just enter the same positive number for each attribute. The choice of the number is irrelevant, it could be 1, 10, 300, 0.007, or any other positive number. If the weights are different for the attributes, then their ratio is all that matters. The weights are local in that they refer only to the attributes in the same subset (i.e., in the same subdivision of an attribute). For example, the weight entered for each of the five MOPs of the OI, Executable, is relative to the weights for the other four MOPs of this OI only. The weight entered for each of the three OIs under an MOE is relative to the weights for the other two OIs of this MOE only. Table 11-3 shows a partial example. The five MOPs under the OI, Executable, and the ten MOPs under the OI, Timely Information, are shown with weights of 1. These are normalized by the tool to 0.20 and 0.10, respectively. The three OIs under MOE 1 are shown with equal weights of 10. These normalize to 0.333.

Table 11-3. Partial example of weights.



	A	B	C	I	K
3	**Code**			**Name**	**Wt**
5				FUNCTIONAL ASSESSMENT	
6	1			MOE 1: Warfighting Information	100
7	1	1		Executable	10
8	1	1	1	Logplan elements	1
9	1	1	2	Sustainment (1 day)	1
10	1	1	3	Critical shortfalls	1
11	1	1	4	Asset information (90%)	1
12	1	1	5	Forecast issues (5 day)	1
13	1	2		Timely Information	10
14	1	2	1	Key information response time	1
15	1	2	2	Sustainment response time	1
16	1	2	3	Schedule response time	1
17	1	2	4	Critical shortfall ID time	1
28	1	2	5	Asset information response time	1
29	1	2	6	Forecast response time	1
20	1	2	7	Drill down time	1
21	1	2	8	Database access & query	1
22	1	2	9	Deployment response time	1
23	1	2	10	Movement update	1
24	1	3		Adaptive	10

Enter the names of the systems under evaluation in Columns L through AQ (a maximum of 32 systems) of row 1. (In the files delivered, unused columns are hidden; use Excel's Unhide command to show these columns.) Enter the assessments for the systems on the attributes in rows 5 through 54, as appropriate. Table 11-4 shows a partial example for the functional assessment of UltraLog 01, UltraLog 02, UltraLog 03, Minimum, and Goal corresponding to the illustration presented in Section 4.4. Notice that entries are made only for attributes at the lowest level in the hierarchy (MOPs). Entries are in units used for the metrics of the attributes. For attributes that have no metric, enter the utilities. The example shown in Table 11-4 corresponds to the entries in Table 4-1 in Section 4.4.

After this information is entered, results can be calculated and displayed. If Excel has been set to recalculate manually, press the F9 function key to calculate all of the outputs. The outputs are described in the remainder of Section 11.3.

11.3.3 Utility curve

The Utility Curve tab provides both tabular and graphical displays of the utility curve for a specified attribute. This is an output supplied by MAUL from the information input under the Input Structure tab (see Section 11.3.2). On the Utility Curve tab, the user specifies the node of interest in the highlighted cell labeled "Select node," and the tool displays the utility curve for that attribute. The node is selected by entering its "Index" number from the Show Node tab, which is described in Section 11.3.4. Be careful to select only a node that has a defined utility curve (with TRUE in column J of the Input Structure tab, which is described in Section 11.3.2.). Selection of any other node will result in an error warning, but will not terminate the program. Simply acknowledge the warning by clicking the OK button and continue with your analysis.

Table 11-4. Partial example of assessments against attributes

	A	B	C	I	L	M	N	O	P
1					Ultra Log 01	Ultra Log 02	Ultra Log 03	Minimum	Goal
2					**1**	**2**	**3**	**4**	**5**
3	**Code**			**Name**	**Sc**				
5				FUNCTIONAL ASSESSMENT					
6	1			MOE 1: Warfighting Information					
7	1	1		Executable					
8	1	1	1	Logplan elements	10%	7%	5%	25%	5%
9	1	1	2	Sustainment (1 day)	30%	10%	6%	25%	5%
10	1	1	3	Critical shortfalls	85%	90%	92%	75%	90%
11	1	1	4	Asset information (90%)	5	2	0.75	12	0.5
12	1	1	5	Forecast issues (5 day)	55%	70%	85%	50%	75%
13	1	2		Timely Information					
14	1	2	1	Key information response time	2	1	0.5	3	0.5
15	1	2	2	Sustainment response time	2.5	1.5	0.75	3	0.5
16	1	2	3	Schedule response time	1.5	1	0.5	3	0.5
17	1	2	4	Critical shortfall ID time	1	0.75	0.5	3	0.5
28	1	2	5	Asset information response time	2	1	0.5	3	0.5
29	1	2	6	Forecast response time	45	30	10	60	5
20	1	2	7	Drill down time	45	30	10	90	5
21	1	2	8	Database access & query	2	1.5	0.5	3	0.5
22	1	2	9	Deployment response time	100	80	65	120	60
23	1	2	10	Movement update	8	4	1.5	10	1
24	1	3		Adaptive					

Selection of a node causes the tool to retrieve and display information for the selected attribute. This display includes: the outline code and name of the attribute (which were entered in columns A-G and I of the Input Structure tab), a tabular listing of input levels and their corresponding utilities in the form of Table 5-1 (as entered in columns AR-BB and BC-BM of the Input Structure tab), and a graph of the utility curve in the form of Figure 4-2.

11.3.4 Show node

The Show Node tab provides a tabular display of the MAU analysis output calculation at any specified node in the hierarchy. It also contains a complete list of all nodes in the hierarchy. The Show Node tab lists all of the nodes in the hierarchy of attributes. It provides an index number that is used to select a node on this tab for display of the MAU analysis results, on the Utility Curve tab (see Section 11.3.3) to select a utility curve for display, on the Local Wt. Sensitivity tab to select a node's weight for the sensitivity analysis, and on the Cum. Wt. Sensitivity tab to select a node's weight for the sensitivity analysis (see Section 11.3.7).

For example, to display the top-level results, enter node 0 in the Select node cell. This displays the top-level evaluation, such as that shown in Table 11-5.

To display the evaluation under MOE 2: Operate Effectively, enter its node number, 23 in the Select node cell. This displays the result shown in Table 11-6.

Table 11-5. Results display for node 0 of the functional assessment.

	WT	CUMWT	Ultra Log 01	Ultra Log 02	Ultra Log 03	Minimum	Goal
FUNCTIONAL ASSESSMENT	1.0000	1.0000	(25)	61	93	0	100
1 MOE1: Warfighting Information	0.5000	0.5000	(20)	56	94	0	100
2 MOE 2: Operate Effectively	0.5000	0.5000	(30)	67	91	0	100

Table 11-6. Results display for node 23, MOE 2: Operate effectively.

	WT	CUMWT	Ultra Log 01	Ultra Log 02	Ultra Log 03	Minimum	Goal
2 MOE 2: Operate Effectively	0.5000	0.5000	(30)	67	91	0	100
2.1 Interoperable	0.3333	0.1667	(6)	60	92	0	100
2.2 Reliable	0.3333	0.1667	7	48	82	0	100
2.3 User friendly	0.3333	0.1667	76	92	100	0	100

The display shows the attribute chosen and its subattributes, the local and cumulative weight of each attribute, and the MAU evaluations of the systems against each attribute. At the bottom level in the hierarchy, this display shows the utility corresponding to the assessment (see Table 11-4). MAUL assumes that utility is piece-wise linear between points specified on the utility curve (see Table 11-2). If the assessment is at a value that is worse than the zero-utility level, a utility of 0 is shown in parentheses and in red, and the 0 is used in all calculations. MAUL highlights all calculations that depend on that value by showing the utility in parentheses (e.g., as shown for Ultra Log 01 in Table 11-6).

11.3.5 Discrimination

The Discrimination tab shows a tabular display of the sensitivity analysis that compares the utilities of two specified systems. Enter the numbers of the systems to be compared (the number of a system is in row 2 of the Input Structure tab, under the system's name). The entry is made of the Favored Alternative # and the Compared Alternative #. The discrimination analysis displays the contributors to the difference in the overall MAU evaluation of the Favored Alternative to the Compared Alternative. This analysis takes into account the differences in performance of the systems against the attributes and the weights of the attributes in the MAU analysis. The designation of one system as Favored and the other as Compared does not affect the operation of the software, only the order of the display of the result (e.g., a system might even be compared with itself; the program will still operate). The ordered list starts with the attributes that favor the Favored Alternative (positive numbers) and ends with the attributes that favor the Compared Alternative (negative numbers). Attributes that favor neither (i.e., the performance is the same or the attribute's weight is zero) are also shown (as zeros). The "wtd dif" column shows the contribution of the individual attribute to the overall difference in evaluations. The "cum" column is the sum of the "wtd dif" column through each row. Table 4-5 in Section 4.6 shows the discrimination analysis with the Goal (alternative #5) as the Favored Alternative and UltraLog 02 (alternative # 2) as the Compared Alternative.

11.3.6 Cumulative weight sort

The Cum. Wt. Sort tab provides a tabular list of the attributes in order of their cumulative weights in the analysis. The cumulative weight of an attribute is the product of the weights along the path leading to the node in the hierarchy of attributes. For example, the cumulative weight for the attribute, Logplan Elements, in the functional assessment is the weight for MOE 1 times the weight for Executable times the weight for Logplan Elements: (0.50)(0.33)(0.20) = 0.033. An examination of the cumulative weight sort can lead to refinements in the assessed weights in an analysis.

11.3.7 Local and cumulative weight sensitivities

The Local Wt. Sensitivity tab provides a graphical display of the sensitivity of the overall evaluation of systems to changes in the local weight given to a specified attribute. The Cum. Wt. Sensitivity tab provides a graphical display of the sensitivity of the evaluation of systems to changes in the cumulative weight given to a specified attribute. These displays show how the overall MAU evaluations of the systems change as the local or cumulative weight assigned to a specified attribute changes from 0.00 (none of the weight) to 1.00 (all of the weight). Specify a node to analyze by entering its node index number.

This is one of the most common types of sensitivity analysis usually performed on an MAU analysis. It is most useful when comparing different designs to determine how sensitive the identification of the best design is to the particular weights in the MAU (i.e., whether the choice of the best design is sensitive or insensitive to particular weight assignments). This type of sensitivity analysis is less important and less interesting when comparing the changing evaluation of a single system over time, such as the illustrative examples used in this report.

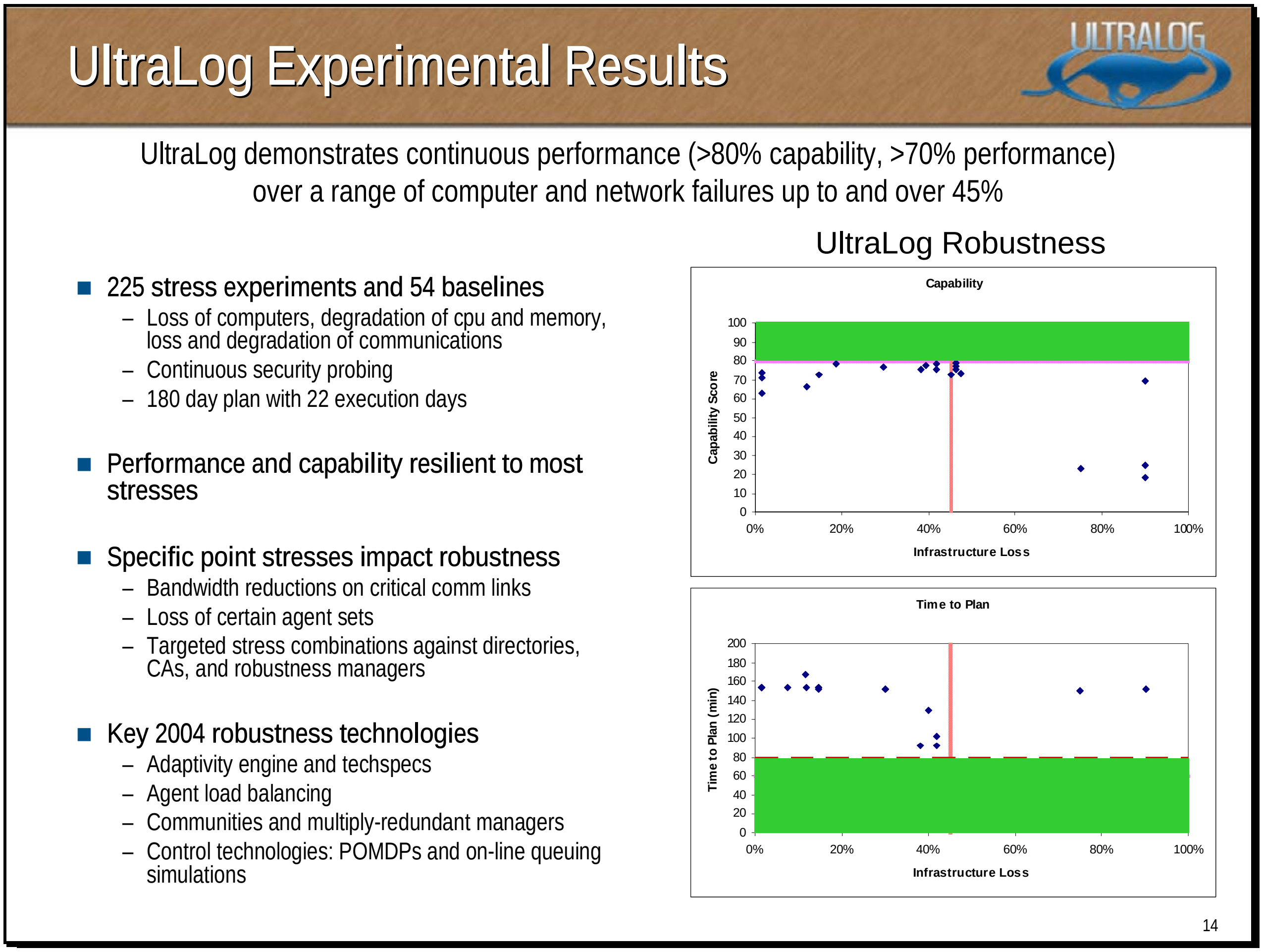
12. IMPACTS OF WORK

The MAU and infrastructure loss approaches, metrics, and calculators developed and discussed in this report were transitioned to and employed by other UltraLog contractors. Their uses and the associated results were then reviewed as an ongoing task by DSA.

The MAU hierarchy of attributes, the assessed utility functions, and the assessed attribute swing weights described in this report were transitioned to S/TDC both in the form of algorithms and as the spreadsheet-based system described in the previous Section. S/TDC then incorporated the algorithms and methods and data into the UltraLog software used to generate the experimental data.

The metric developed for infrastructure loss was embedded in a number of calculators provided to the UltraLog team. The infrastructure loss metric was employed to select experiments to test the survivability claim. Additionally, it was programmed by other contractors into the UltraLog software used to generate the experimental data.

As an example of the use of both MAU and infrastructure loss metrics and methods, Figure 12-1 depicts some of the results described in June of 2004 at the Military Operations Research Society meeting in Monterey, California (Ulvila et al., 2004). The upper graph depicts the relationship between infrastructure loss and Capability score from the MAU model. The lower graph plots the same experiments, but displays the relationship between infrastructure loss and Time to Plan.



UltraLog Experimental Results

UltraLog demonstrates continuous performance (>80% capability, >70% performance) over a range of computer and network failures up to and over 45%

- 225 stress experiments and 54 baselines
  - Loss of computers, degradation of cpu and memory, loss and degradation of communications
  - Continuous security probing
  - 180 day plan with 22 execution days
- Performance and capability resilient to most stresses
- Specific point stresses impact robustness
  - Bandwidth reductions on critical comm links
  - Loss of certain agent sets
  - Targeted stress combinations against directories, CAs, and robustness managers
- Key 2004 robustness technologies
  - Adaptivity engine and techspecs
  - Agent load balancing
  - Communities and multiply-redundant managers
  - Control technologies: POMDPs and on-line queuing simulations

Figure 12-1. UltraLog Experimental Results, June 2004.

In general, the sensitivity analysis methods developed and supplied with the MAU approaches and with the MAUT spreadsheet were not employed. This was because the UltraLog system displayed such a high level of survivability that most of the MAU attribute scores did not differ at all in the stressed and baseline runs.

Finally, the MAU and infrastructure loss metrics developed and discussed in this report formed the basis for using the experimental data to make the survivability claim.

REFERENCES

Brown, R. and Ulvila, J. The role of decision analysis in international nuclear safeguards. In P. Humphreys, O. Svenson, and A. Vari, *Analyzing and aiding decision processes*. Amsterdam: North-Holland, 1983, pp. 91-103.

Edwards, W. Use of multiattribute utility measurement for social decision making. In Bell, D., Keeney, R., and Raiffa, H. (Eds.), *Conflicting objectives in decisions.* NY: Wiley, 1977.

Jensen, F. *An introduction to Bayesian networks.* NY: Springer, 1996.

Keeney, R. *Value focused thinking.* Cambridge, MA: Harvard University Press, 1992.

Keeney, R., and Raiffa, H. *Decisions with multiple objectives.* NY: Wiley, 1976.

Pigaty, L. Chaos, performance, and capability. (Working Paper) Los Alamos Technical Associates, Inc., 17 July 2001.

Pratt, J., Raiffa, H., and Schlaifer, R. *Introduction to statistical decision theory.* Cambridge, MA: MIT Press, 1995.

Saydjari, O. S., Wood, B. and Bouchard, J. UltraLog Active Assessments Plan. Albuquerque, NM: SRI International, May 2001.

Ulvila, J. Ideas on environmental chaos. (Working Paper) Vienna, VA: Decision Science Associates, Inc., 13 July 2001.

Ulvila, J., Boone, J., Chinnis, J., and Gaffney, J. *Multiattribute Utility Analysis for UltraLog.* (Technical Report 02-2) Vienna, VA: Decision Science Associates, Inc., 2001(a).

Ulvila, J., Boone, J., Gaffney, J., Notargiacomo, L., and Welke, S. *Framework for information assurance attributes and metrics.* (Technical Report 01-1) Vienna, VA: Decision Science Associates, Inc., 2001(b).

Ulvila, J. and Chinnis, J. Decision analysis for R&D resource management. In D. Kocaoglu, *Management of R&D and engineering*. Amsterdam: North-Holland, 1992, pp. 143-162.

Ulvila, J., Chinnis, J., and Dyson, M. Multiattribute Utility Theory for UltraLog. Monterey, California: Military Operations Research Society meeting. June, 2004.

UltraLog Functional Assessment (7/17/01). Los Alamos Technical Associates, Inc., 2001.

APPENDIX A: ULTRALOG FUNCTIONAL ASSESSMENT (7/17/01)

MOE/OI/ MOP	Description	Metrics
MOE 1	Does the system provide useful warfighting information.	
OI 1-1	As survivability measures are implemented or as the system is attacked, does this system produce an executable logplan and instill user confidence	
MOP 1-1-1	How accurate, compared to ground truth, are key logplan elements such as customer wait time and order and shipment status.	G: average error<5% 1 day out, <25% 5 days out W: average error> 25% 1 day out
MOP 1-1-2	How accurately can the system estimate gross resupply and other sustainment requirements.	G: average error<5% 1 day out, <25% 5 days out W: average error> 25% 1 day out
MOP 1-1-3	How accurately can the system identify critical logistics shortfalls based on operational planning factors, CINC critical item and special requirements lists, and time phased requirements.	G: 95% of shortfalls identified accurately W: 75% of shortfalls identified accurately
MOP 1-1-4	How accurately can the system provide real-time information on the location and condition of sustaining assets, including assets in production, in-storage and in-transit, regardless of location, including applicable contractors.	G: 90% accuracy ½ hour after information is available from source W: 70% accuracy 12 hours after information is available from source
MOP 1-1-5	How accurately can the system predict and forecast upcoming issues in the logistics chain, to include customer wait time for goods and services.	G: 90% accuracy 1 day out, 75% 5 days out W: 90% accuracy 1 day out, 80% 2 days out, 70% 3 days out, 50% 5 days out
OI 1-2	As survivability measures are implemented or as the system is attacked, does the system provide timely information.	
MOP 1-2-1	How long does it take the system to respond to queries about key logplan elements such as customer wait time and order and shipment status.	G: 95% query response < 30 sec. W: 5% query response >3 mins
MOP 1-2-2	How long does it take the system to respond to queries about gross resupply and other sustainment requirements.	G: 95% query response < 30 sec. W: 5% query response >3 mins
MOP 1-2-3	How long does it take the system to respond to queries involving assets, schedules, projected and actual consumption, sourcing, current and historical cycle times for procurement, repair and delivery.	G: 95% query response < 30 sec. W: 5% query response >3 mins
MOP 1-2-4	How long does it take the system to identify critical logistics shortfalls based on operational planning factors, CINC critical item and special requirements lists, and time phased requirements.	G: 95% query response < 30 sec. W: 5% query response >3 mins
MOP 1-2-5	How long does it take for the system to provide real-time information on the location and condition of sustaining assets, including assets in-production, in-storage and in-transit, regardless of location, including applicable contractors.	G: 95% query response < 30 sec. W: 5% query response >3 mins
MOP 1-2-6	How long does it take the system to predict and forecast	G: 95%<5 mins.

	upcoming issues in the logistics chain, to include customer wait time for goods and services.	W: 5%>60 mins.
MOP 1-2-7	How long does it take to perform drill down operations for data on displayed objects.	G: 95%<5 sec. (after the 1st) W: 5%>90 sec. (after 1st)
MOP 1-2-8	How long does it take the system to access and query asset, readiness, and personnel databases.	G: 95% query response < 30 sec. W: 5% query response >3 mins
MOP 1-2-9	How long does it take the system to produce level six deployment data for a contingency.	G: Level 6 in 60 mins. W: Level 6 in 120 mins.
MOP 1-2-10	How frequently is on-line movement planning and execution status updated.	G: Update in 1 min. W: Update in 10 mins.
OI 1-3	As survivability measures are implemented or as the system is attacked, does the system remain adaptive.	
MOP 1-3-1	What is the system lag time for dynamic replanning.	G: 15 mins. W: 30 mins.
MOP 1-3-2	How long does it take the system to develop fully sourced and time-phased logistics course-of-action (COA) evaluation products with alternatives. This includes vulnerability assessments, sensitivity analyses, and draft execution documents.	G: 15 mins. W: 30 mins
MOP 1-3-3	What is the impact of wartime volumes of data on the system accuracy and responsiveness.	G: retain 90% accy, 80% res W: retain 50% accy, 50% res
MOE 2	Does the system operate effectively.	
OI 2-1	As survivability measures are implemented or as the system is attacked, does the system remain interoperable with the existing communications architecture.	
MOP 2-1-1	Does the system provide for real-time, secure communications and data exchange interoperability for logistics support entities and command and control functions.	G: 95% real-time, secure interoperability W: 80% real-time, secure interoperability
MOP 2-1-2	Is the system interoperable with existing local communications networks.	G: Interoperable with 100% W: Interoperable with 95%
MOP 2-1-3	Is the system interoperable with existing databases, management servers and engines.	G: Interoperable with 100% W: Interoperable with 95%
MOP 2-1-4	Does the system integrate and coordinate with joint and common logistics support entities and interface with the logistics support structure outside theater boundaries; e.g., national, intermediate, inter-theater, including contractor and allied logistics providers, during both the planning and execution phases of operations.	G: With 95% W: With 80%
MOP 2-1-5	Can the system access GCSS-compliant databases.	G: Yes; W: No
OI 2-2	As survivability measures are implemented or as the system is attacked, does the system remain mission ready and operating.	
MOP 2-2-1	What is the incidence of fault, failure, and malfunction.	G: < 2 failures to access major databases per day W: > 5 failures to access major databases per day
MOP 2-2-2	What is the time to restore the system to its full operating state after fault, failure, or malfunction.	G: 80% restored in 5 mins., 99% in 1 hr.

		W: 80% restored in 15 mins, 99% in 2 hrs
MOP 2-2-3	To what degree did system availability affect user ability to meet mission needs.	4-point scale by expert opinion
OI 2-3	As survivability measures are implemented or as the system is attacked, does the system remain user friendly.	
MOP 2-3-1	Do the user interfaces retain a similar look and feel.	4-point scale
MOP 2-3-2	Do the user interfaces remain intuitive to a trained logistics operator.	4-point scale
MOP 2-3-3	Does the system retain the ability to integrate, visualize and present logistics information.	4-point scale
MOP 2-3-4	Does the system continue to allow execution monitoring of logplan accomplishments compared to objectives at the user desktop PC level.	G: on 90% of items W: on 70% of items
MOP 2-3-5	Does the user retain the ability to readily incorporate system products into staff estimates and briefs.	G: “readily” W: “with great difficulty”
MOP 2-3-6	Does the user retain the ability to quickly integrate system information into user C4I operational environment.	G: “quickly” W: “with long delay”